

---

## USERS MANUAL

---

# Model 2045

# Polynomial

# Waveform Synthesizer



# **ANALOGIC. ■**



---

## **IMPORTANT**

### *Analogic Data Precision Purchaser Registration*

Thank you for purchasing a Data Precision product. It's our objective to provide customer support that equals the quality of the instrument.

Please complete and return this postage-paid registration form immediately, so that we may keep you apprised of revisions to this operating manual, as well as system upgrades, modifications, troubleshooting information, and application notes.

This form should be completed *in addition to* your product warranty card.

---

Name \_\_\_\_\_ Title \_\_\_\_\_

Purchaser/Primary User \_\_\_\_\_

System Purchased \_\_\_\_\_ Date of Purchase \_\_\_\_\_

Company \_\_\_\_\_ Mail Stop \_\_\_\_\_

Group or Division \_\_\_\_\_ Phone (     ) \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Application(s) \_\_\_\_\_

*(please be as specific as possible)*

Other Data Precision equipment you presently use \_\_\_\_\_

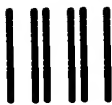
Application(s) \_\_\_\_\_

Comments or Suggestions: \_\_\_\_\_

---

---

---



---

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT #290 DANVERS, MA

---

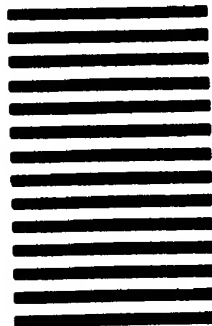
POSTAGE WILL BE PAID BY:

**ANALOGIC. ■**



8 Centennial Drive  
Peabody, MA 01960  
Attn.: Sales Dept.

No Postage  
Stamp Necessary  
If Mailed  
In The  
United States



## Table Of Contents

<b><u>Section</u></b>	<b><u>Topic</u></b>
A .....	Introduction
B.....	The Instrument
C.....	Operation (OP) Codes
D .....	Standard Function Modes
E.....	Polynomial (Arbitrary Waveform) Mode
F .....	Modified Mode
G .....	Remote Programming and Control
H .....	Remote Mnemonics
I.....	Scope Draw
J.....	Tablet Draw
K .....	Upload/Download
L.....	Recovery from Errors
M.....	Specifications
General Index	



# INTRODUCTION

The model 2045 arbitrary waveform generator from Data Precision brings a new realm of capabilities to the waveform synthesis field. A maximum clock rate of 800MHz, waveform resolution to 8 bits and waveform memory of 512K points makes the 2045 suitable for a variety of applications.

## Waveform Outputs

The 2045 has both a high level and a low level output. The high (A) level (5.0V p-p into 50 ohms, 10.0V p-p into infinite load) output has attenuator and user adjustable offset circuits and a bandwidth which is typically 200MHz. The low (B) level (1.0V p-p into 50 ohms, 2.0V p-p into infinite load) output has a simpler output stage without the attenuator or offset circuits, but with a wider bandwidth, typically 1GHz.

## Front Panel (*Figure A-1*)

This shows the 2000 series front panel(s). An examination of this drawing will provide you with a good preview of their features.

## Rear Panel (*Figure A-2*)

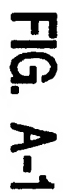
This shows the rear panel of the 2040/2045. Refer to section B for detailed information about rear panel functions.

## Internal View (*Figure A-3*)

This is a view looking down on the 2045 with the top cover removed, showing where the batteries for the non-volatile memory are located and how to raise the card cage. See section B for more information about the batteries.

## Block Diagram

A functional block diagram is provided for those interested in the basic architecture of the 2045.





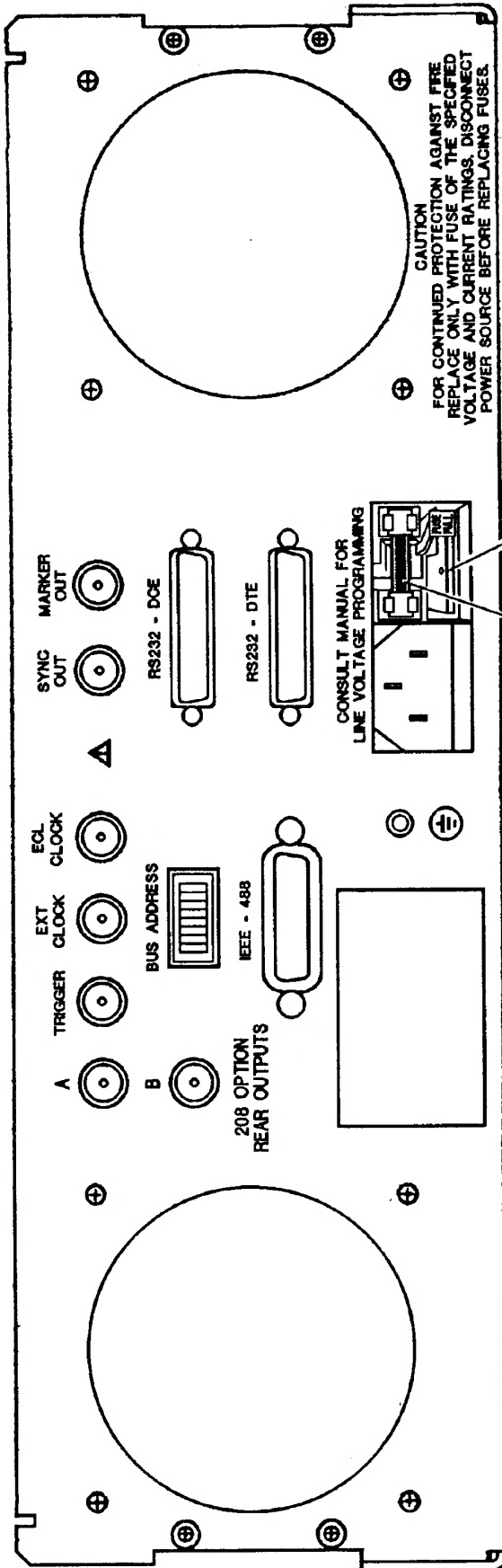


FIG. A-2

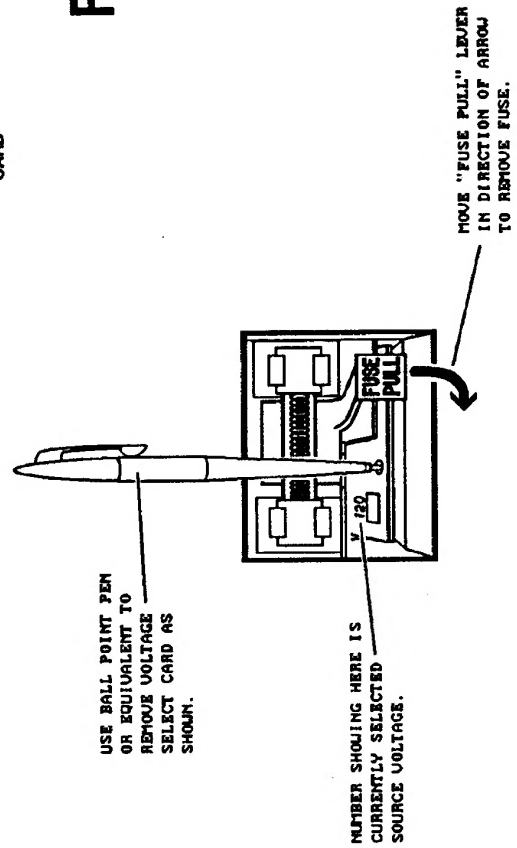


FIG. A-3

TOP VIEW  
WITH COVER  
REMOVED

LOOSEN TWENTY SCREWS TO TILT CARD CAGE UP FOR CIRCUIT BOARD ACCESS.  
CAUTION: IF INSTRUMENT IS OPERATED WITH CARD CAGE UP, EXTERNAL FAN SHOULD BE USED TO MAINTAIN AIRFLOW THROUGH CARD CAGE.

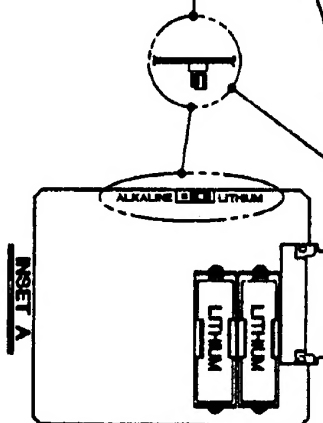
BATTERIES FOR  
NON-VOLATILE  
MEMORY. SEE  
INSET A.

INPUT LINE FUSE  
(ACCESS FROM  
REAR PANEL)

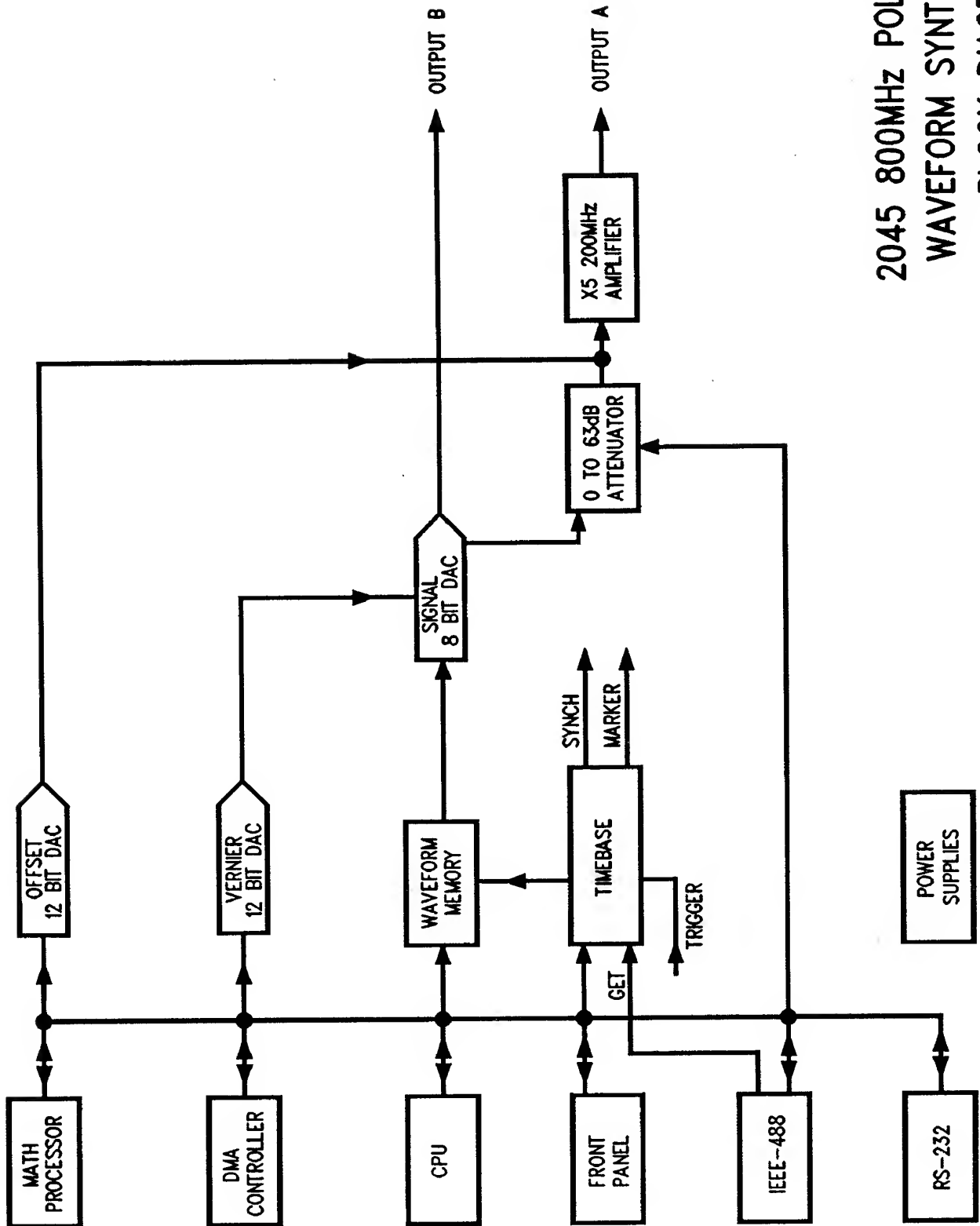
● LITHIUM POSITION SHOWN  
● ALKALINE POSITION OPPOSITE  
(SEE INSET A)

#### BATTERY USE

- AA LITHIUM BATTERIES ARE SUPPLIED.
- REPLACE WITH AA LITHIUM OR AA ALKALINE BATTERIES ONLY.
- DO NOT USE CARBON BATTERIES.
- RECOMMENDED MANUFACTURER FOR LITHIUM BATTERIES: SANYO
- RECOMMENDED MANUFACTURER FOR ALKALINE BATTERIES: AM BURELL, AM EVEREADY, OR EQUIVALENT.



# 2045 800MHz POLYNOMIAL WAVEFORM SYNTHESIZER BLOCK DIAGRAM





# THE INSTRUMENT

## 1. OPERATION OVERVIEW - Signal, Waveform, Data Sources

There are several sources from which the 2045 can formulate its output signal:

1. Standard Function Modes; by which classical wave shapes such as square, pulse, sine, triangle, sawtooth, etc. are readily generated.
2. Polynomial Mode; by which waveshapes are generated from user supplied mathematical expressions of the form  $Y = f(t)$ . User expressions of this form can be named and stored in internal non-volatile memory.
3. Real World waveforms can be captured (digitized) on any Data Precision 6000 Series Waveform Analyzer and downloaded to the 2045 for replication without the need for an additional controller.
4. A predetermined digital pattern data can be downloaded from any intelligent host (via IEEE-488 or RS-232) for analog generation by the 2045.
5. Scope Draw mode; where waveform data may be modified while the waveform is displayed on a scope.
6. Tablet Draw mode; where a drawing tablet is used to draw a waveshape while it is displayed on a scope.

## 2. SETUP AND POWER UP

### Connecting The Power Source

Before connecting the instrument (2045) to an A.C. power source be sure that the power source (115 VAC or 220 VAC) matches the voltage selection on the rear panel of the instrument and verify that the power source is of the third wire grounded type.

### Voltage Select Card & Fuse

Figure A-2 shows the line voltage selection card detail and describes input line fuse removal. Before replacing the fuse, determine the cause of the excessive current so that it does not recur. Replace the fuse only with the same type and rating.

### Batteries

Figure A-3 details the location and replacement for the non-volatile batteries. These batteries should last several years. If they do need replacing, a power-up Low Battery message will be displayed. Memory data should not be lost while changing the batteries provided that they are not left out of their holders for more than ten minutes or so. Note that if you change the type of batteries (Lithium or Alkaline), the battery jumper must be set to the proper position.

## Power Up and Memory Test

**CAUTION:** If your load device (connected to a signal out connector) could be upset or damaged by a voltage that is within the output range of the 2045, your load should be connected after the 2045 is turned ON. As is the general case with microprocessor-controlled signal generators, the output state cannot be established until the processor has initialized. Therefore, instantaneous voltages during the power-on transient could exist anywhere within the output stage voltage capability. Additionally, RUN [R/S] should not be executed until you have ENTERED the waveform that is to be RUN, i.e., Standard Function, Polynomial Function, DownLoaded Function, etc.

When the unit is turned on, a waveform memory read/write test is performed. A message will appear advising you that the test is in progress. The message will also tell you that pressing any key will abort the test, if you don't wish to wait for the test to complete. If you are operating the unit remotely, anything you send will be interpreted as an interrupt and the memory test will be terminated. Also, when operating remotely, if you want the test to run to completion and advise you when the test is complete, a serial poll will tell you when the memory test status bit goes low and when the Listen bit is high. See "SRQs and Status Bits" in section G for more information.

## Hardware/Firmware Version and Special Features

The message that appears after turn-on and initialization describes the model and firmware revision number. Special features can be reviewed by using the key sequence [2nd] [OP] [7] [ENTER] . The down cursor arrow is used to view additional message lines when the prompt (more) is shown. Use [RESTORE] to exit the special feature listing mode.

## Help

A quick understanding of the use and function of any or all of the front panel keys can be obtained by using the built in HELP feature. Pressing the HELP key places the instrument into the *Help Text Mode*. While in this mode, pressing any front panel key will bring to the display a brief explanation of that key. To EXIT the HELP mode press the HELP key again. There is also a HELP feature associated with the OP CODE routines. See the OP codes section for more information.

## Assistance and Service

This instrument has been designed for ease of operation and long life. If, however, you require either applications or service assistance, please consult an Analogic/Data Precision local representative or the factory in Peabody, Massachusetts. For applications assistance, call 1-800-343-8333. For service assistance, call 1-800-237-2200.

### 3. CONNECTIONS (Other Than A.C. Power)

#### FRONT PANEL (Refer to front panel view shown in figure A-1)

##### Output A and Output B

These are the Analog Signal output connectors. Output A is the high level output and has a range of  $\pm 5V$  into 50 ohms or  $\pm 10V$  into an infinite load with user adjustable attenuator and offset circuits. Output B is the low level output and has a range of  $\pm 0.5V$  into 50 ohms or  $\pm 1V$  into an infinite load with software offset and no attenuator.

#### REAR PANEL (Refer to rear panel view shown in figure A-2)

##### TRIGGER

This is the connector that is used to either trigger or gate the 2045 when it is operating in other than the Free-Run mode. The active threshold can be set to either zero-crossing or plus 1.5 volts by means of the OP 4 configuration set-up (refer to OP CODE Section). Refer to the Triggering Modes descriptions in either the STANDARD FUNCTION or POLY Sections of this manual.

##### EXT CLOCK

This is a logic level signal input that provides an output data clock to the 2045 when the clock source configuration is set to EXTERNAL in the OP 4 set-up routine. The logic threshold is configurable as either zero crossing or plus 1.5 volts, also selected in OP 4 (See the OP CODE Section).

##### ECL CLOCK

This is an ECL level only input, with a maximum input range of  $+0.5V$ ,  $-4.0V$ . It provides an output data clock to the 2045 when the CLK SOURCE is set to ECL in the OP 4 menu.

##### SYNC OUT

This is a TTL active high logic signal that occurs at the beginning of each output waveform pass. It is particularly useful for synchronizing an oscilloscope for output waveform viewing.

##### MARKER OUT

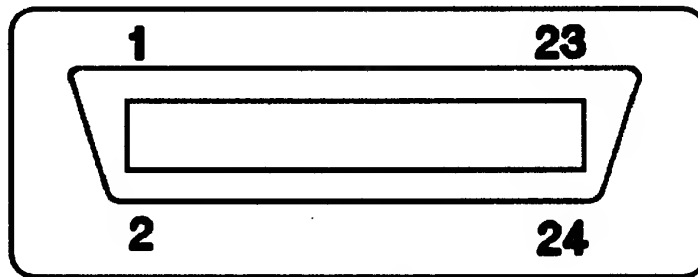
This is a TTL active high logic signal that can be positioned to occur at any point during the output waveform duration. It is set by either an attribute statement in POLY mode or directly set as a menu field parameter in either STANDARD FUNCTION mode or MODIFIED mode.

**IEEE-488**

This is a standard IEEE-488 connector that is used for remote (BUS) control of the 2045. Refer to Figure B-1 for a detailed pin-out of this connector.

The 2045 has the capability of handling the following IEEE-488 interface functions:

	<u>Interface Function</u>	<u>Code</u>
1.	Source Handshake	SH1
2.	Acceptor Handshake	AH1
3.	Talker	T6
4.	Listener	L4
5.	Service Request	SR1
6.	Remote Local	RL1
7.	Parallel Poll	PP0
8.	Device Clear	DC1
9.	Device Trigger	DT1
10.	Controller	C0
11.	Driver Electronics	E2



**Figure B-1 IEEE-488 Pin Numbering**

*Table BT-1: IEEE-488 Pin Assignments:*

<u>Pin</u>	<u>Description</u>	<u>Pin</u>	<u>Description</u>
1	DI01	13	RNFD
2	DI05	14	GND
3	DI02	15	NDAC
4	DI06	16	GND
5	DI03	17	IFC
6	DI07	18	GND
7	DI04	19	SRQ
8	DI08	20	GND
9	EOI	21	ATN
10	REN	22	GND
11	DAV	23	SYSGND
12	GND	24	GND



## RS-232 (DCE) and RS-232 (DTE)

These are RS-232 remote control connectors that are used for Data Console and Data Terminal operation when the RS-232 option is installed. Refer to Tables BT-2 and BT-3 for detailed pin-out assignment on these connectors. When connecting an IBM PC (or compatible) asynchronous communications interface (serial card) to the 2045, the 25 pin DCE port is connected directly to the 25 pin DTE port on the IBM PC interface card. However, in order to connect the DTE port on the 2045 to the DTE port on the IBM you must fabricate a NULL MODEM interconnect cable as follows:

<u>2045 Instrument End</u>	<u>P.C. Serial Card End</u>
Pin 1 (Gnd)	Pin 1 (Gnd)
Pin 7 (Gnd)	Pin 7 (Gnd)
Pin 2 (TXD)	Pin 3 (RXD)
Pin 3 (RXD)	Pin 2 (TXD)

If the Serial Card utilizes DTR-DSR handshake:

Pin 5 (CTS)	Pin 20 (DTR)
Pin 20 (DTR)	Pin 6 (DSR)

If the Serial Card utilizes RTS-CTS handshake:

Pin 5 (CTS)	Pin 4 (RTS)
Pin 20 (DTR)	Pin 5 (CTS)

Figure B-2

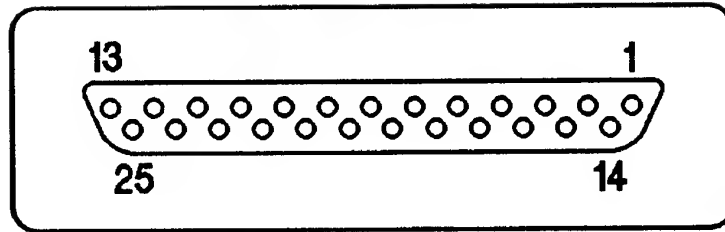
RS-232 Pin  
Numbering

Table BT-2: RS232 DCE Pin Assignments:

Pin	Description	Pin	Description
1	GND	14	
2	TXD	15	
3	RXD	16	
4		17	
5	CTS	18	
6	PULL UP	19	
7	GND	20	DTR
8	PULL UP	21	
9	GND	22	
10	GND	23	
11		24	
12		25	
13			

TXD: Transmitted Data, Input

RXD: Received Data, Output

CTS: Clear To Send, Output

DTR: Data Terminal Ready, Input

Table BT-3: RS232 DTE Pin Assignments:

Pin	Description	Pin	Description
1	GND	14	
2	TXD	15	
3	RXD	16	
4		17	
5	CTS	18	
6		19	
7	GND	20	DTR
8		21	
9	GND	22	
10	GND	23	
11		24	
12		25	
13			

TXD: Transmitted Data, Output

RXD: Received Data, Input

CTS: Clear To Send, Input

DTR: Data Terminal Ready, Output

## SETTING DEFAULT COMMUNICATION PARAMETERS

### LISTEN SRQ, COMMAND DEVICE and BUS ADDRESS

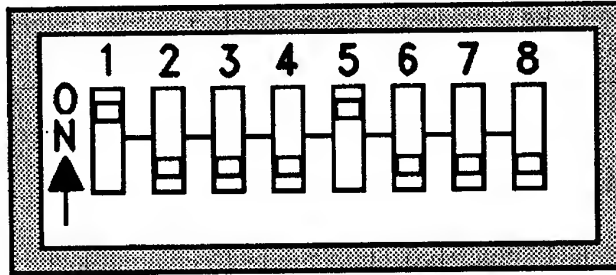


Figure B-3 Rear Panel Dip Switches

Figure B - 3 shows the rear panel slide switch assembly that sets the default Listen SRQ mode, Command Device and IEEE-488 bus address. Switch position 1 determines the default Listen SRQ mode. Switch positions 2 & 3 are used to select the command device as either IEEE-488 or RS- 232 (DTE[Ser 1] or DCE [Ser 0] connector). Switch positions 4, 5, 6, 7 & 8 are used to specify the IEEE-488 bus address in a binary coded manner where position 8 is the LSB. Listen SRQ mode, Command Device selection and IEEE-488 Bus Address defaults can be changed by remote software control or by using the appropriate OP configuration menu (ref. OP Code and Remote Sections).

Note: U = UP (ON), D = DOWN (OFF); refer to fig. B-3

#### LISTEN SRQ MODE:

##### Switch Number 1 Listen Service Request (SRQ)

U	enabled
D	disabled

#### COMMAND (CMD) DEVICE:

<u>Switch Number 2</u>	<u>3</u>	<u>DEVICE</u>
D	D	IEEE-488
D	U	RS-232 (DCE) Ser 0
U	D	RS-232 (DTE) Ser 1
U	U	IEEE-488

#### GPIB BUS ADDRESS:

<u>Switch Number</u>					
4	5	6	7	8	ADDR
D	D	D	D	D	00
D	D	D	D	U	01
D	D	D	U	D	02
D	D	D	U	U	03
D	D	U	D	D	04
D	D	U	D	U	05
D	D	U	U	D	06
D	D	U	U	U	07
D	U	D	D	D	08
D	U	D	D	U	09
D	U	D	U	D	10

<u>Switch Number</u>					
4	5	6	7	8	ADDR
D	U	D	U	U	11
D	U	U	D	D	12
D	U	U	D	U	13
D	U	U	U	D	14
D	U	U	U	U	15
U	D	D	D	D	16
U	D	D	D	U	17
U	D	D	U	D	18
U	D	D	U	U	19
U	D	U	D	D	20
U	D	U	D	U	21

<u>Switch Number</u>					
4	5	6	7	8	ADDR
U	D	U	U	D	22
U	D	U	U	U	23
U	U	D	D	D	24
U	U	D	D	U	25
U	U	D	U	D	26
U	U	D	U	U	27
U	U	U	D	D	28
U	U	U	D	U	29
U	U	U	U	D	30
U	U	U	U	U	31



## OPERATION (OP) CODES

OP CODES are used to implement and/or change system and I/O (input / output) configurations.

OP CODES are invoked by the key sequence 2nd, OP, #, #, #, #, ENTER; where the #'s indicate the OP Code number which will be four digits or less.

A listing of OP Codes in your machine can be obtained by executing the key sequence 2nd, OP, HELP. The display will then show you the first in a list of OP codes with a brief explanation of that code and "(more)" in the lower right hand corner of the display. This means that depressing the DOWN ARROW key will bring up another OP CODE and its explanation. When you get the last OP CODE listing, the display will tell you to "Press <HELP> or <RESTORE> to return." This means that pressing either of these keys will exit the OP CODE HELP routine. If you press ENTER while an OP code appears on the display, That OP code will be invoked just as if you pressed 2nd, OPxxx, ENTER.

The following explanations show the screen display that you should expect to get when you call the OP CODE which shows the default parameters and values. Menu field selection, parameter value changing, up line paging, etc, are all consistent with the editing explanations given in the STANDARD FUNCTION section of this manual.

To change menu values or attributes: If there is a limited number of selections available for a particular menu field, the selection can be changed, or toggled between choices, by using either the UP ARROW or the DOWN ARROW cursor keys. In the case of a widely variable numerical value, as in TARGET PTS, simply enter the numeric value. A flashing value indicates that the change has not been executed; use the ENTER key to execute the new value.

**IMPORTANT:** No menu field value or attribute changes are actually implemented until the ENTER key is depressed. This means that if you make a change and exit the OP CODE routine with the RESTORE key, the change will not be realized if you have failed to ENTER the new value.

### OP 0 Scope Draw Mode

---

OP 0 is used to enter Scope Draw mode. See the Scope Draw section (I) for more information.

### OP 4 2045 General Parameters

---

The display shows:

FUNCTION OUT = A	TARGET PTS = 1000
TRIG LEVEL = 1.5V	CLK SOURCE = INTERNAL
CLK LEVEL = 1.5V	IDLE VOLTAGE = 0V
CMD DEVICE = GPIB	

FUNCTION OUT selections: A or B

Definition: Selecting one of the output channels allows you to define the output of that channel, in either standard function or poly mode. The channel not selected, although affected by the selected channel, is not under full user control, and the relationship between the two channels is not symmetrical.

Generally, outputs A and B are intended to be used separately and not simultaneously. Although A and B originate from the same DAC, A has its own attenuator, offset and amplifier circuits. When A is selected, any adjustments to amplitude on A will affect the amplitude output of B by reducing it 0 to 1 dB from full scale. If you slew down the amplitude of A in standard function mode while monitoring B, you will notice a repeating pattern of B going down and back up by this 0 to 1 dB range. Also note that when A is selected, the B output expresses the peak to peak waveform variation over its range. As a result, the offset of such a waveform may be in error.

With the foregoing in mind, when A is selected, B remains available for use simultaneously if the small deviations in amplitude and possible offset error are not a problem for you. Waveshape and frequency will be identical. However, when B is selected, the attenuator in A is set to maximum, effectively reducing the output of A close to zero, so A is not available for practical use with B selected.

**TARGET PTS:** (Change by keying in desired numeric value)

Maximum: 524288      Minimum: 64

Note: The minimum given above is the smallest number that can be entered in OP 4. Depending on the requirements of the waveform, the actual number of points computed can be as low as 2.

Definition: Waveforms generated in the Standard Function Mode use this value as a target number of points per cycle for the waveform being generated. In the Polynomial (Arbitrary) mode, the number of data points in the output waveform will be this number or less, unless a clock rate is specified that requires more points. Increasing the TARGET PTS value results in higher horizontal resolution (finer grain waveform) at the cost of longer computation time.

**TRIG LEVEL:** Selection: 1.5V, 0V

Definition: External Trigger Level Threshold in Volts

**CLK SOURCE:** Selection: INTERNAL, +EXTERNAL, -EXTERNAL, ECL

Definition: Output Waveform Data Point Clock is derived either INTERNALLY or EXTERNALLY. EXT CLOCK and ECL CLOCK are BNC connectors on the rear panel, so marked.

**CLK LEVEL:** Selection: 1.5V, 0V

Definition: External Clock Input Trigger Threshold in Volts.

**IDLE VOLTAGE:** Selection: 0V

This field tells you that when in an idling state, the output is driven to 0 volts. There is no other selection.

**CMD DEVICE:** Selection: GPIB, SER0, SER1, NONE

Definition: This menu selection determines the I/O Command Device. The rear panel selector switch assembly determines the default CMD Device selection. Refer to figure B-3 for further clarification on the rear panel selector switch assembly.

## **OP 7      Special Features Listing**

---

The key sequence [OP] [7] [ENTER] will bring to the display a listing of these features. At the end of each option listing, the display shows "more" which means that the DOWN cursor key should be pressed to advance the listing. This routine will also report the amount of non-volatile RAM space available. This available memory report is an

indication of the remaining (unused) RAM space that can be used to store poly waveform descriptors. However, a small portion (less than 4K) of this reported space is reserved for segmented output memory operation in the direct data download mode (refer to the Remote Programming Section). Pressing the RESTORE key will exit the OP 7 utility.

## OP 232 RS-232 I/O Parameters

---

Op 232 is used to configure the RS-232 I/O parameters. The two RS-232 ports, DCE (SER0) and DTE (SER1), have independent parameter configurations. The selection in the first field (PORT) tells you which port is currently being configured.

**IMPORTANT:** Please note that this menu is used for port configuration only. To select a port for remote command input, refer to the OP4 menu.

The display menu fields and default values for OP 232 are as follows:

PORT = SER0:	BAUD RATE = 9600
STOP BITS = 1	PARITY = OFF
DUPLEX = FULL	LINE LENGTH = 80
TERMINATOR = <CR>	PROMPT = ON

**PORT:**

Selections: SER0, SER1:

**BAUD RATE:**

Selections: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200

**STOP BITS:**

Selections: 1, 2

**PARITY:**

Selections: OFF, ODD, EVEN

**DUPLEX:**

Selections: FULL, HALF

**LINE LENGTH:**

Selections: 80, 132

**TERMINATOR:**

Selections: [CR], [LF], NULL

**PROMPT:**

Selections: ON, OFF

**Note:** When using either the RS-232 or IEEE-488 communications a DEVICE TIMEOUT message indicates that communications could not be established during a 5 second timeout period.

## OP 488 IEEE-488 (GPIB) I/O Parameters

---

OP 488 is used to configure the IEEE-488 (GPIB) I/O parameters. As with OP232, OP488 is used only for port configuration. The command device for remote command input must be selected in the OP4 menu.

The display shows:

Bus address = 16	EOI out = ON
GET response = NONE	Talk SRQ = OFF
Listen SRQ = OFF	Error SRQ = OFF
Fast I/O = OFF	

Bus address:

Range: 0 to 31

Definition: IEEE-488 address assignment for this unit

Note: Reassignment of the bus address in this OP Code does not affect the power-on default address. The default address is determined by the rear panel DIP switch setting as described in figure B - 3.

EOI out:

Selections: ON = End Or Identify word is expected on receipt and supplied on transmission.

OFF = End Or Identify word is optional on receipt and not supplied on transmission.

GET (IEEE-488 Group Execution Trigger) response:

Selections: NONE = GET not recognized

TRIGGER = Trigger previously loaded output memory at receipt of GET

Talk SRQ:

Selections: OFF = No Service Request sent in Talk mode

ON = Service Request is activated in Talk mode

Listen SRQ:

Selections: OFF = Service Request not sent in Listen mode

ON = Service Request is sent when 2045 is ready to listen

Note: The Listen SRQ default is determined by the rear panel switch assembly shown in Figure B - 3.

Error SRQ:

Selections: OFF = Service request not sent when an error is detected.

ON = Service request is sent when an error is detected.

Fast I/O:

Selections: OFF = IEEE-488 communication speed is LOW (normal)

ON = IEEE-488 communication speed is HIGH



## OP 1510 Tablet Draw Mode

---

OP 1510 is used to enter Tablet Draw mode. See the Tablet Draw section (J) for more information.

## OP 6000 Captured Waveform Retrieval From DATA 6000

---

OP 6000 lets you copy a waveform from the directory of any Analogic/Data Precision 6000 series waveform analyzer (including the D6100) to the 2045 so it can be regenerated. After entering OP 6000, the copying process (captured waveform retrieval) may proceed as shown below.

In the first display window, the 2045 is asking you to confirm that the Data 6000 Waveform Analyzer has a GPIB Bus Address of 15. If this is not correct, the Data 6000 Address can be entered by stepping up or down with the UP or DOWN cursor keys. The range of available addresses is 00 through 31.

- 1) DATA 6000 bus address = 15  
Step to correct address then <ENTER>

Pressing ENTER now results in the second display which indicates that the 2045 is establishing communications and reading the Data 6000 File Directory.

- 2) Reading directory . . .  
Step to desired file then <ENTER>.

If communications with the Data 6000 cannot be established (within 5 seconds), the third display (device timeout) will appear. In this event, check the D6000 to verify the command device in the DIR/PROG menu and the selected port in the OPT 'Line Format' menu are both GPIB. Verify that the D6000 bus address entered in OP 6000 agrees with the setting on the D6000. Since each device on an IEEE-488 bus must have a unique address, check the 2045 bus address in OP488 and make sure it is not the same as the D6000.

- 3) Device timeout

When you have established communications with the D6000, the display will show the directory contents of the D6000 as it might appear below. The first file name will be flashing, which means it is currently selected. Pressing the ENTER key will tell the 2045 to download this file. If that is not the file you want, change the flashing selection left or right and up or down with the CURSOR arrow keys. If there are more file names than can be contained on the two display lines, automatic line paging will occur so that you can scroll through the entire D6000 directory.

- 4)
 

BUF.A1	BUF.A2	BUF.A3	MAGCA1
AVEGCA2	DIFFA3		

Pressing the ENTER key now will transfer the selected file to the 2045. While this file transfer is taking place, the display will appear as shown below in view 5:

- 5)
 

Reading file . . .
Please wait

In the sixth display view, the 2045 has downloaded the Data 6000 waveform file name BUF.A1 containing (in this example) 12545 data points.

- 6)                    BUF.A1 endpoint = 12545 Step to change.  
                      Press <ENTER> or <RESTORE> to exit.

At this point, the number of data points to be reproduced by the 2045 can be reduced by changing the end point to some number less than what is shown. This truncation of the waveform end point may be desired if, for instance, the downloaded waveform is not an even multiple of periods and a continuous (free run) 2045 output is desired with no wraparound discontinuity.

The new end point value can be slewed down to by holding the down arrow key or it can be keyed in from the number keypad. If you hold down the FAST key while slewing the endpoint, the value will jump to the nearest block boundary of 64 rather than stepping by one point. If you key it in at the keypad, the new value will flash, at which point you should press ENTER to execute the change. If you discover that you have slewed down the end point value too far, or keyed in too low a value and you have not yet pressed the ENTER key, you can slew back up or key in a higher value without the losing the affected data. Keep in mind that once you press ENTER, the end point is set and any data that you bypassed is now gone.

If you now press ENTER or RESTORE to exit OP 6000, the retrieved file will remain in output memory. If you press the DIR key, the OMEM (Output Memory) part of the directory listing will now show "D6000.". This data will remain in output memory until you delete it. See section E for an explanation of the directory.

## **OP 6001     Change End Point of Retrieved Waveform**

---

**Definition:** To change the waveform end point after execution of retrieval. This is used to resolve an end point discontinuity for a continuous reconstruction.

**Range:** Maximum = 524288  
              Minimum = 64

OP 6001 actually recalls the last step of OP 6000:

                      BUF.A1 end point = 12545 Step to change.  
                      Press <ENTER> or <RESTORE> to exit.

**Note:** OP 6001 can only be executed after a file has been retrieved with OP 6000 and while the file is still in output memory. If the "D6000." listing in the directory is deleted, then it is gone from memory and entering OP 6001 will yield only a blank display. The only way to recall the file in this event is to retrieve it again with OP 6000.

## **OP 9999 Erase Non-Volatile Memory**

---

**Definition:** Used to delete entire contents of 2045 non-volatile memory.

The Internal Directory of the 2045 is further described in the POLY mode section of this manual where selective file deletions are explained. In general, files should be individually deleted rather than using this (OP 9999) bulk erase method. This bulk erase operation does not affect the portion of non-volatile RAM that is reserved for segmented output memory control. This reserved memory is further described in OP 7 of this Section and in the Remote Programming Section of this manual. If directory difficulties are experienced due to having lost battery back-up power, this bulk erase (OP 9999) routine should always be used to reinitialize and reallocate the RAM space.

When OP 9999 is invoked by pressing ENTER, the display shows:

This will delete all files saved.  
Do you want to continue? <YES> or <NO>

If complete erasure is your intention, depress the YES key; otherwise escape by pressing NO.

## **FACTORY DIAGNOSTIC CODES:**

---

There are other OP CODES that are reserved for Factory Diagnostic routines that are used during the manufacturing cycle. Some of these OP codes are used in the maintenance and calibration manual. These OP Codes are not described here and are mentioned only in the event that an erroneous entry of an intended OP Code yields something that is not recognizable to the operator. In this case abort the extraneous OP Code by pressing RESTORE.



# STANDARD FUNCTION MODES

There are four standard function modes:

## 1. SQUARE WAVE

## 2. SINE WAVE

## 3. TRIANGLE WAVE

## 4. NOISE

Any one of the four standard function modes is invoked by depressing the appropriate key showing a symbol corresponding to a square wave, sine wave, triangle wave, or labeled NOISE. When one of the four standard modes is selected, the LED indicator next to the mode key will illuminate and remain on as long as the selected mode is invoked.

Each standard function mode will bring menu fields to the display to enable the operator's selection of certain waveform parameters relevant to the invoked function. Each of these menu fields is preset with a default value or condition. For example, frequency is preset to 100.000 KHz. One of the fields will contain a cursor to indicate the active field; that is, the field that is subject to change. The select cursor can be sequentially jumped to the next field to the right by depressing the key marked FIELD or by simultaneously depressing the key marked FAST and the right arrow key. The cursor can be jumped to the next field to the left by simultaneously depressing FAST and the left arrow key. The cursor can be moved under particular digits within the numerical value of the current (selected) field by depressing either the left arrow or the right arrow keys without FAST.

There are two methods available for changing the numerical value of a particular field parameter. They are "slew" and "direct overwrite".

To slew a value, place the cursor under the digit to be slewed and press the up arrow key to increment the digit or the down arrow key to decrement the digit. Carry to more significant digits is automatic. If the run LED is lit, the slew will be considerably slowed down by an internal recomputation process between each successive value. To avoid this, depress the R/S (run - stop) key to disable the output until the displayed value is satisfactory, then activate the output (R/S) at which time only one recompute process will be invoked. Or, with the RUN LED on (output active), depress the FAST key simultaneously with the up or down arrow key. In the latter case, the displayed value will change but the output function will not be updated until the FAST key is released.

To overwrite a numerical value, simply enter the desired value using the numerical keypad. The overwrite can be started with the cursor under any digit in the current field value. In the direct overwrite method, the use of a decimal point is optional and the use of suffix notation such as M (Meg), K (Kilo), m (milli), u (micro) is optional. The suffix notation is located as a shift on the numerical keypad and is accessed by preceding the numeral key with the blue "2nd" key. For example, [K] is located over the numeral 1 key, therefore to enter 2.67 KHz, one would use the following key sequence: [2] [.] [6] [7] [2nd] [1]. When the desired value entry is complete, you MUST press ENTER to execute the value change and the RUN LED (R/S key) must be on if you expect to see the new value at the output.

After a numerical value change is made by either of the above methods and a recompute is invoked by run ON (preceded by ENTER in the case of a direct overwrite), the displayed value may change to the nearest realizable value, if the requested value cannot be exactly synthesized. The new displayed value is the actual value being provided by the 2045.

When a field value has been changed but not yet executed, the entire field value will be blinking as a reminder that ENTER is required to execute the changed value.

To change field parameters that are non-numeric such as trigger mode options or noise on/off; use the up or down arrow keys to cycle through the available alternatives.

## **Format - Menu Field Alternatives**

Some fields can be expressed in alternate forms such as FREQUENCY/PERIOD or DUTY CYCLE/PULSE WIDTH. Fields that have an alternate units form are identified with a colon (:) instead of the equals sign (=) that is used for fixed format field parameters. The displayed field format is changed by pressing the FORMAT key while the select cursor is in the field to be changed. Each depression of the FORMAT key toggles between the alternate format forms. Since the selection is actually expressing the same parameter in alternate units, a change to the numeric value in one format automatically changes the value of the alternate form.

For example, if the frequency is 100 KHz, depressing FORMAT will cause the display to show PER:10u (period is 10 uS). If the period is changed to 50u (50 microseconds) a FORMAT key toggle will cause the display to show FREQ:20K. Note that the parameter value can be specified (entered at the numeric keypad) or slewed (using the arrow keys) in either format. The fields that have format alternatives are shown in the menu field illustrations which follow, as distinguished by the colon (:) notation. Field values and/or conditions are carried between standard function modes, except for DLY (delay). That is, if the operator changes the frequency value in the square wave mode and then changes the mode to a sine wave, the frequency will remain at its last setting (the one specified in the previous mode).

## **Standard Function Modes - Overview**

In the following view of the SQUARE WAVE menu fields, the first two lines indicate the fields that initially appear in the two line active display area. There are four lines making up the whole menu, and these will scroll up one line at a time as the field select cursor is advanced beyond the current line end. For example, when the cursor is stepped right to AMP and then to OFST and then to FILT the lines beginning with FILT and NOISE will occupy the two display lines available for viewing. The lines also wrap; that is, when the field select cursor is advanced past the last field, the first two lines will reappear. This works the same way with somewhat different menus for the other Standard Function modes.

## 1. SQUARE WAVE MODE:

The menu fields and default values for the SQUARE WAVE MODE are as follows:

FREQ:100.000K	AMP:1.00027	OFST:0.00000
FILT = NONE	MARK:0.00000	TRIGM = FREE RUN
NOISE = ON	NAMP = 0.00000	NBW = 200KHz
DLY = 0.00000	DUTY:50.0000%	

The FORMAT key will change the following fields:

FREQ changes to PER:10u

AMP & OFST change simultaneously to HIGH:1.00027      LOW:-1.00027

MARK changes to MPTS:0

DUTY changes to PLSW:5.00000u

## 2. SINE WAVE MODE:

The menu fields and default values for the SINE WAVE MODE are the same as the square wave mode except for the last two fields:

PHS = 0.00000      SYM = 50.0000%

DLY (Delay) in square wave mode relates to PHS (Phase) in sine wave mode. DUTY (Duty Cycle) in square wave mode relates to SYM (Symmetry) in sine wave mode. Note, however, when changing from SQUARE to SINE, the Duty Cycle value is carried over to Symmetry, but the Delay value is not carried over to Phase. This is equally true if changing from SINE to SQUARE. The reason is that Duty Cycle and Symmetry use the same units (percent) whereas Delay and Phase use different units (seconds and cycles, respectively)

## 3. TRIANGLE WAVE MODE:

The menu fields and default values for the TRIANGLE WAVE MODE are the same as the square wave mode except for the last field:

SYM = 50.0000%

Symmetry in TRIANGLE mode is the same as Symmetry in SINE mode, measured in percent and related to Duty Cycle in SQUARE mode.

### Alternate Trigger Mode Format:

In all three of the standard function modes above, if you set GET = TRIG in the OP488 menu, this will change the TRIGM field to GET = START or STOP. This alternate form is not affected by the FORMAT key, as indicated by the equal (=) sign.

#### 4. NOISE MODE:

The menu fields and default values for the NOISE MODE are the same as the three noise fields from the wave shape modes above with two additional parameters:

NOISE = ON    NAMP = 0.00000    NBW = 200KHz  
NCLK = 10u    NSEED = 1

The clock rate for the NOISE mode (NCLK) determines the available 3dB noise bandwidths (NBW). There are approximately 15 bandwidths available for each clock rate. SEED is a starting point in a string of randomly generated noise points. For more information, see the SEED command in the mnemonic reference section.

#### Noise Generator

A software psuedorandom noise generator is used to generate values that have approximately normal probability distribution. These values are loaded into output memory with any desired signal components and subsequently made available as output at the set clock rate. It should be noted that the noise will repeat at a rate depending on the number of points used in the computation. The signal may be repeated in the memory so that different noise values occur over each signal cycle.

A software filter is incorporated to allow approximately single pole low pass filtering of the random noise. The frequency spectrum of the output is given by:

$$SD(f) = \frac{V_o}{\sqrt{ENBW}} * ABS \left( \frac{1}{D - [(D-1)e^{-j2\pi fT}]} * \frac{SIN(\pi fT)}{(\pi fT)} \right)$$

Where:

SD(f) = Spectral Density of output, in Volts / (Hz)<sup>1/2</sup>

T = Period of output clock

V<sub>o</sub> = Output RMS voltage

D = A filter parameter having values of 1, 2, 4, 8...16384

f = Frequency

ENBW = Effective Noise Bandwidth from table below

For large values of D the spectrum approaches that of a single pole low pass filter.

A table follows that enables calculation of all possible 3dB bandwidths obtainable with a given clock. The 2045 does the calculation and allows selection of the bandwidth directly. Also shown in the table is the effective noise bandwidth which is equal to the bandwidth of an ideal rectangular low pass filter having the same low frequency spectral density and the same total noise power as the output.

D	3dB BW * T	ENBW * T	D	3dB BW * T	ENBW * T
1	443e-3	0.5	128	125e-5	196e-5
2	110e-3	116e-3	256	623e-6	978e-6
4	458e-4	713e-4	512	311e-6	489e-6
8	213e-4	333e-4	1024	156e-6	244e-6
16	103e-4	161e-4	2048	777e-7	122e-6
32	505e-5	793e-5	4096	388e-7	610e-7
64	251e-5	394e-5	8192	194e-7	305e-7
			16384	971e-8	153e-7



*As an example:*

The widest 3dB bandwidth achievable is  $0.443/1.25\text{e-}9 = 354 \text{ MHz}$  and the Effective Noise Bandwidth is  $0.5/1.25\text{e-}9 = 400\text{MHz}$ .

Note that this bandwidth is further reduced by the inherent bandwidth of the A and B outputs being typically 200 and 1000 MHz respectively.

Continuing the example, for an RMS output of 0.5V, the low frequency spectral density is:

$$V_o \div (\text{NBW})^{1/2} = 0.5 \div (400\text{e}6)^{1/2} = 25\mu\text{volts} / (\text{Hz})^{1/2}$$

## Standard Function Menu Fields - Detail

### FREQUENCY (FREQ:) *INTERNAL CLOCK*

Units: cycles per second (Hertz) are implied unless other prefix is specified

Minimum: 284.212nHz

Maximum: 400MHz

All Standard Functions are computed by placing one or more cycles of the desired function into Output Memory (OMEM) utilizing the highest available Clock Rate and the nearest possible number of points per cycle to the TARGET PTS ("target" number of points) set in OP 4. The 2045 will round the number of points up or down if necessary to get as close as it can to the called for frequency. With a very small number of points, the difference in resolution can be more noticeable, but larger numbers of points make this negligible.

Note: For a Standard Function, vertical (amplitude) resolution does not improve above TARGET PTS = 780. This is because of the limitations of the 8 bit DAC. However, horizontal (frequency) resolution will continue to improve as larger numbers of points are computed, allowing you to obtain progressively smaller increments of frequency shift.

### FREQUENCY (FREQ:) *EXTERNAL CLOCK*

When CLOCK (CLK) is set to EXTERNAL (EXT) in the OP 4 configuration mode, the display shows FREQ = EXT since the output frequency is a function of the external clock rate being used. Since the TARGET PTS value is used to describe one waveform cycle in the Standard Function Mode, the output frequency when using external clock is the external clock frequency  $\div$  the actual number of points computed. The actual number of points computed may not be exactly the same as the TARGET PTS selected, but it will be as close as possible. The default TARGET PTS value is 1000 and it can be changed in the OP 4 configuration menu to any value from 64 up to 524288.

### AMPLITUDE (AMP):

Units: Volts PEAK into 50 ohms

PEAK-TO-PEAK (into 50 ohms) will be two times indicated value

Unloaded PEAK output will also be two times indicated value

Maximum (Output A):  $\pm 2.5$  Volts into 50 ohms ( $\pm 5$  V open circuit)

Maximum (Output B):  $\pm 0.5$  Volt into 50 ohms ( $\pm 1$  V open circuit)

Minimum: 0 Volts

**FILTER (FILT = )**

Number: 2

Type: Passive (lumped Constant) modified Bessel, 3 Pole, low pass Selectable 3 Db frequencies: 2 MHz, 20 MHz and NONE

Roll-off: 18 dB/Octave

Note: NONE results in the inherent output bandwidth which is 160 MHz minimum (200 MHz typical) which allows a 2.2nS maximum rise time. Filter selection does not alter output impedance. The selection of a filter (except NONE) affects amplitude flatness and settling time as expected due to the filters transfer function relative to the signal frequency spectrum.

**MARKER TIME (MARK: )**

Units: Seconds

Measured From: Leading edge of SYNC output pulse (rear panel, TTL)

Minimum: Zero (i.e., coincident with the SYNC output)

Maximum: 2 to the 21st (approx 2 million) waveform points times the clock period being used.

Note: The MARKER pulse is a fixed 32 clock periods long and can be placed only in 32 clock period increments.

**MARKER POINTS (MPTS: )**

Units: Number of Waveform Data Points. In this alternate MARKER notation, the marker is set at a particular waveform point by counting data points.

Minimum: Zero data points (i.e., coincident with the SYNC output)

**NOTES ON THE MARKER OUTPUT:**

Output: Rear Panel; BNC connector

Level: TTL output capable of driving 50 ohm load to ground.

Polarity: Positive leading edge

Duration: 32 clock periods

**TRIGGER MODE (TRIGM = )**

Input: Rear panel; labeled TRIGGER

Allowable Input Voltage Range:  $\pm 30$  V

Input Impedance: 10 K $\Omega$  (< 50pf)

Trigger Threshold Voltage: either 0 or + 1.5 volt; selected by setup configuration (OP 4) as described in the OPERATION CODES Section of this manual.

Selections (When OP488 GET response = NONE):

<u>Remote Command</u>	<u>Trigger Mode</u>
TRIG = 0 .....	FREE RUN
1 .....	+ START
2 .....	- START
3 .....	+ GATED
4 .....	- GATED
5 .....	+ STOP
6 .....	- STOP
7 .....	+ TO +
8 .....	- TO -

If the OP 488 parameter GET response = TRIG, then the IEEE-488 bus command GET will force a "trigger event" in the 2045. In this case, trigger mode (TRIGM becomes GET ) has two selections which are different from the trigger modes when GET response = NONE.

Selections (When OP488 GET response = TRIG):

<u>Remote Command</u>	<u>Trigger Mode</u>
TRIG = 9 .....	START
10 .....	STOP

*TRIGGER MODE selections (TRIGM = ) when GET response = NONE : Detail*

**FREE RUN:** Continuous output (not triggered or gated) with a SYNC out pulse at the start of each record. A record typically contains 64 cycles of the waveform. This ensures that a record ends on a 64 point boundary.

+ **START:** Triggered by the positive going TRIGGER IN transition, or by depressing the MANUAL TRIG key. Once triggered, the output is continuous.

– **START:** Triggered by the negative going TRIGGER IN transition, or by depressing the MANUAL TRIG key. Once triggered, the output is continuous.

+ **GATED:** Output Function is active while Trigger Input voltage exceeds the trigger threshold, or while the MANUAL TRIG key is held depressed.

– **GATED:** Output Function is active while Trigger In voltage is less than the trigger threshold voltage, or while the MANUAL TRIG key is held depressed.

+ **STOP:** Output Function is active until a positive going TRIGGER IN transition is experienced, or until the MANUAL TRIG key is depressed. Output is started by ENTER RUN key sequence.

– **STOP:** Output Function is active until a negative going TRIGGER IN transition is experienced, or until the MANUAL TRIG key is depressed. Output is started by ENTER RUN key sequence.

+ **TO +:** Output Function STARTS on positive going TRIGGER IN edge and STOPS on next positive going TRIGGER IN edge or, starts on first depression of the MANUAL TRIG key and stops on the next depression of the MANUAL TRIG key.

– **TO –:** Output Function STARTS on negative going TRIGGER IN edge and STOPS on next negative going TRIGGER IN edge or, starts on the first depression of the MANUAL TRIG key and stops on the the next depression of the MANUAL TRIG key.

*TRIGGER MODE selections (GET = ) when GET response = TRIG : Detail*

**START:** A trigger will enable continuous output. The trigger can be generated by a GET command over the IEEE-488 bus or by pressing the MANUAL TRIG key.

**STOP:** A trigger will disable output. The trigger can be generated by a GET command over the IEEE-488 bus or by pressing the MANUAL TRIG key.

**NOISE (NOISE = )**

Selections: ON, HOLD

ON: Noise is enabled, and will appear at the output with the amplitude defined by NAMP.

HOLD: Noise is disabled.

**NOISE AMPLITUDE (NAMP = )**

Units: Volts rms

In calculation of effective Signal to Noise ratio, the operator should take note of the fact that signal amplitude is expressed in Volts PEAK and noise is expressed in Volts R.M.S.

**NOISE BANDWIDTH (NBW = )**

Selections: There are 15 bandwidths available for each NCLK value selected. The total range is from  $\approx 18\text{Hz}$  to  $\approx 354\text{MHz}$ .

**NOISE CLOCK (NCLK = )**

NCLK sets the clock rate for NOISE mode only. It operates independently of CLK.

Selections: The range here is the same as the regular CLK, from 1.25nS to 515.380S.

**NSEED (NSEED = )**

NSEED is used to select a starting point in a string of noise data. See the NSEED command listing in the Command Reference part of section H for more information.

Selections: NSEED can equal any integer value in the range  $\pm 9999999$ , except NSEED cannot equal zero.

**DELAY (DLY: )**

Units: Seconds

Definition: Time from leading (positive) edge of SYNC output pulse to leading (positive) edge of SQUARE or PULSE wave at the output connector. In the case of TRIANGLE or SAWTOOTH modes, the DELAY time is measured to the center of the leading (positive) slope of the output.

**PHASE (PHS: )**

Units: Cycles

Range: 0 to 1

Definition: Phase shift from the leading (positive) edge of SYNC output to the positive zero crossing of the OUTPUT sinewave in decimal portions of a cycle. For example, 0.33333 would yield one third cycle (120 degrees) phase shift.

Resolution:  $\pm$  one data point

**DUTY CYCLE (DUTY: )**

Units: Percent

Minimum Percentage: ( clock period/waveform period) \* 100

Maximum Percentage: ((waveform period – clock period) / waveform period) \* 100

Note: Clock period can be determined from examination of the MODIFIED mode menu. Waveform period can be determined by pressing the FORMAT key when the (FREQ = ) menu field is selected.

**PERIOD (PER: )**

Units: Seconds

Minimum: 2.5 nanoseconds

Maximum: 3.5185 Megaseconds.

Note: This field (PER = ) is accessed by a FORMAT key toggle when the (FREQ = ) field is selected.

**PULSE WIDTH (PLSW: )**

Units: seconds

Minimum: 1 clock period

Maximum: waveform period minus one 1 clock period

Note: Clock period can be examined in the (CLOCK = ) field of the MODIFIED mode menu.

Note: This field (PLSW = ) is an alternate format for the (DUTY = ) field in SQUARE WAVE mode, selectable with the FORMAT key.

**HIGH LEVEL (HIGH: ) LOW LEVEL (LOW: )**

Units: Volts

Range: 5.0 V p-p (A), 1.0V p-p (B)

Note: The HIGH/LOW fields are accessed by a FORMAT key toggle while either the (AMP = ) or the (OFST = ) field is selected in one of the Standard Function modes. This field nomenclature could be useful in Square wave mode when emulating logic level drive where the HIGH level specifies the positive or logic 1 level, and the LOW level specifies the negative or logic 0 level.

**SYMMETRY (SYM = )**

Units: Percentage

Resolution:  $\pm (1/\text{TARGET PTS}) * 100$ 

Note: Used in the SINE WAVE and TRIANGLE WAVE modes in the same manner as, and with the same characteristics as, DUTY CYCLE (DUTY = ) in the SQUARE WAVE mode. If set to other than 50.0% in the SINE WAVE mode, an asymmetrical sine wave will be produced. If set to other than 50.0% in the TRIANGLE WAVE mode, the Triangle Wave will become asymmetrical and appear as a SAWTOOTH wave. Greater than 50.0% will yield a longer positive slope and proportionately shorter negative slope and less than 50.0% will yield the opposite, a shorter positive slope and a longer negative slope. Frequency will not be affected by adjustments to symmetry.

## POLYNOMIAL (ARBITRARY WAVEFORM) MODE

In this mode, the 2045 acts as an ARB (Arbitrary Waveform Generator) utilizing a waveform descriptor language based on an algebraic format. In this mode the 2045 can be likened to a Video Calculator on which your time domain expression in the form  $Y = f(t)$  is evaluated and generated as a voltage function of time  $V = f(t)$ . This mode provides a very powerful method for the generation of essentially any waveform.

### To enter the "POLY" mode:

If the 2045 is in any of the STANDARD FUNCTION MODES (SFM), depress the key for the currently activated SFM. The SFM light will go out and a blinking cursor will appear in the upper left corner of the display, indicating that you are now in the POLY mode. After power-up (turn-on) and initialization the unit is in the POLY mode even though the display shows the message of introduction. By pressing any key which generates a menu, character or function display, the introductory message will be removed and the appropriate display will appear. If you remove it by pressing any of the cursor arrow keys, the display will go blank except for the flashing cursor.

### An example:

The algebraic form of a sinusoidal waveform is  $V = a \sin(\omega t + \Phi)$  where  $V$  is the instantaneous output voltage,  $\omega$  is the angular rate,  $a$  is the amplitude coefficient,  $t$  is time on the horizontal axis (starting at zero), and  $\Phi$  is the phase offset.

### CYC/RAD mode:

For trigonometric arguments, the 2045 accepts either radians per second (RAD) or cycles per second (CYC). The default (power-on) mode is CYC. When the CYC indicator is lit, trigonometric angular rates ( $\omega$ ) is in cycles per second and phase offset ( $\Phi$ ) is in cycles, i.e., 0.25 cycles relates to 90 degrees. When in the RAD mode (RAD indicator lit) angular rate ( $\omega$ ) and phase offset ( $\Phi$ ), are both in radians. The CYC/RAD mode must be properly selected before the ENTER key is depressed. Unless otherwise specified, the CYC mode will be assumed throughout this section.

### Example continuation:

Let's assume 1 (unity) for the amplitude coefficient and zero for  $\Phi$  with a 1 Megahertz frequency.

1. Turn on the 2045 and observe a short diagnostic routine which will be followed by the message of introduction. If the unit is already on and in a SFM mode such as Square Wave, depress the Square Wave key to exit SFM and enter the POLY mode. A blinking cursor in the left most display position is your indication that you are in the POLY mode.

2. Execute the following KEY SEQUENCE:

[FOR] [1] [2nd] [u] [SPACE] [SIN] [1] [2nd] [M] [x] [T] [)]

3. Observe on the display: "FOR 1u SIN(1M\*T)" which means do for one microsecond the sine of one Megahertz. The engineering notation milli [m], micro [u], Kilo [K], etc. do not imply dimensions such as Volts or Seconds. Rather, the dimensions are understood on the 2045 by virtue of their location within the expression. In this case, micro [u] is taken as microseconds and Mega [M] is taken as MegaCycles per second. The left parenthesis after SIN is automatically inserted. The [space] is optional in that it only provides a more readable expression upon viewing the display. We also started with an evaluation time, in this case one microsecond.

4. Connect the SYNC OUT on the 2045 to trigger a dual trace oscilloscope. The SYNC pulse occurs once for every waveform record, and is useful as a stable trigger. Connect the 2045 output to the Vertical input of the oscilloscope, through the 50 ohm terminator (supplied with the 2045) with appropriately set vertical and horizontal scale factors to observe a 1 Megahertz sine wave on the other scope trace. Note that the sine wave is not yet being generated by the 2045.

5. Press the R/S key to enable the output (note that the RUN LED lights) and then the ENTER key. Or, press the ENTER key followed by the R/S key. In either case, the output must be in RUN mode and the polynomial must be entered before the waveform will be computed. Note that the display says "computing 800 points". The default target number of horizontal points to be computed is 1000. In this example, 800 is the nearest value to 1000 the 2045 can compute with the polynomial numbers involved. See the OP 4 command in the OP CODE Section if you wish to change the target number of data points to be computed).

6. Observe a one Megahertz sine wave of about one volt peak amplitude on the oscilloscope. The actual amplitude and several other parameters can be checked by pressing the RESTORE key. Press the RESTORE key again to return to the polynomial. You will also note that the zero crossing (zero phase) is coincident with the positive edge of the SYNC OUT pulse and that the waveform is continuous even though we had specified one microsecond. This is because the default 2045 triggering mode is FREE RUN so that the specified one microsecond is being repeated, continuously. Other triggering modes will be discussed under TRIGGERING and in the SFM Sections of this manual.

At this point, perhaps you are wondering about the horizontal level in between each cycle. This is because the 2045 can only generate waveforms in blocks of 64 points, where the total number of points in the record is a multiple of 64. If a waveform is defined (as in the example) that violates this condition, then the last point of the defined waveform is extended to the next 64 point boundary. In this example, only 800 points were computed because the minimum clock period is 1.25nS and 800 points produces the programmed 1uS duration. 800 not being a multiple of 64, if you wish to get a continuous sine function then simply increasing the duration to 2uS will allow computation of 1600 points which is divisible by 64. See "Data Block Length Restrictions" later in this section.

## Editing:

Editing is easy. Let's now change the discontinuous 1 MHz sine wave expression to a smoothly continuous sine wave. We could clear the entire display with the key strokes [2nd] [CLR] and re-key the entire expression, but for the sake of editing practice let's change the previous expression, which should still be on the display as FOR 1u SIN(1M\*T) with a blinking cursor.

1. Use the LEFT ARROW to step the cursor left until it is under the 1 in 1u. Press the CE key. Now key in 2 and note that the display now shows: FOR 2u (SIN(1M\*T)).

2. Now press [ENTER] and observe the "Computing 1600 points" message followed by the appearance of our revised output on the oscilloscope. Since the number of points computed (1600) is a multiple of 64, the waveform is now smoothly continuous. Of course, if all you want is a simple sine wave, going through this relationship calculation is not really necessary because any sine wave can be generated in Standard Function Mode by a different process where 64 cycles of the function are computed, which ensures that the total record will end on a 64 point boundary. The purpose of this exercise is to show how POLY mode works and how to use it.

## DIRECTORY

### Polynomials: ( Store [STO], Recall [RCL] and Delete [DEL] )

If you would like to store the example expression for future recall you simply give it a name before it is entered. The name must begin with an ALPHA character and be less than nine characters long followed by an equal sign. Let's name this one F33. To do this, we edit the display to read:

"F33 = FOR 2u SIN(1M\*T)"

Now press [ENTER]. Or, instead of editing to affix the file name we could have keyed in [STO] [F] [3] [3] [ENTER]. In either case, the waveform is now stored in permanent (non-volatile) memory. To verify this, turn the 2045 power OFF and back ON. Now depress the directory key [DIR]. There will be one or more expression names on the display, one of which will be F33, and one of them will be flashing. If F33 is not the flashing one, press either the LEFT or RIGHT (UP/DOWN to jump directory lines) arrow keys until F33 is selected (flashing). Now depress the recall [RCL] key. You will note that your expression is recalled to the display. If you activate RUN and press [ENTER] you will regenerate the F33 function.

There are two methods to DELETE an expression. In the first, press DIR, select (cause to flash) the desired expression name and press DEL. The display will ask "Are you sure?" at which point pressing the YES key will execute the deletion. Try this with F33. Verify that you have deleted it by pressing directory DIR and noting that it is no longer listed. Now key in our sample expression and name it F34, press ENTER and press DIR to verify that it has been stored. Now delete it by the second method which is simply the key sequence [DEL] [F] [3] [4]. As in the previous deletion method the display will ask "Are you sure?". If you recall an expression by file name, and change it in any way followed by ENTER in order to execute the changed expression, the display will ask if you wish to replace the old expression with the new since you now have two expressions with the same name. If you do not wish to overwrite the previous expression, either change the name of the modified one or remove its name. Remember, an expression with no name can be executed, but it is not automatically stored. You can construct alpha sequences to form mnemonics or words as file names by pressing keys as LOG, SIN, TAN, etc, and editing out characters with the CE key. Of course, over the remote control bus you can dictate alphanumeric strings that start with an ALPHA character and have no more than eight characters total. To erase the entire memory (delete all files) refer to OP 9999 in the OP CODE Section of this manual.

### Polynomial Expressions and Directory Size:

An executable expression can contain up to 1280 display character locations, which includes file name, spaces, attributes and all other characters that occupy a display location. Note that the display shows up to 80 of these characters at a time with auto scrolling to view the rest. The non-volatile memory can store up to 78K bytes. Each file (expression) requires 20 bytes of directory overhead and each display character requires one byte. Therefore, to store one expression containing 30 display characters we would occupy 50 bytes of non-volatile memory. Note that there is no limit to the number of files (expressions); except that the directory overhead (20 bytes/file) would occupy all of the 78K available space at 3900 files. In realistic terms, considering an average file (expression) length to be 60 characters, the number of stored expressions could be up to 975. Note that some (up to 4K) of the remaining, unused, RAM reported in OP 7 and in the external Directory Dump must be reserved for segmented memory operation, whether or not such segmentation is being used (refer to the Remote Programming Section).



## Directory Listings, Memory (NVR and OMEM) and File Types

There are two separate and distinct memory banks available to you, NVR and OMEM.

### Non-Volatile RAM (NVR)

NVR is for storage of POLY expressions and for internal CPU variable and stack space.

In the foregoing discussion on Directory Store [STO], Recall [RCL] and Delete [DEL] we were dealing with polynomial expressions being stored in the Non-Volatile memory. Data point files (waveforms) can also be stored in OMEM (Output Memory) and manipulated with the same Recall [RCL] and Delete [DEL] commands.

### Output Memory Volatile Ram (OMEM)

OMEM is the memory space that contains the waveform datapoints that actually make up waveforms. The size of OMEM is 512K data points. Up to 16 OMEM files can reside in the directory at one time.

OMEM (data point) files can be distinguished from NVR (polynomial expression) files in that they are appended with a dot [.]. OMEM files are, however, not stored with the Store [STO] command. They are automatically created and stored in one of the following forms:

1. **Direct Data Download** - if not sent with a name, it will be named in the directory as "DATA." (See Remote Programming Section).
2. **Standard Functions** - named in the directory as "STDFN." (See Standard Function section).
3. **Polynomial Data** - named in the directory as "POLY.".
4. **OP 6000 retrieved Data** - named in the directory as "D6000." (See OP Codes section).
5. **Upload/Download Data** - if not sent with a name, it will be named in the directory as "DATA." (See Upload/Download section).

All of the above OMEM files can be recalled [RCL] and RUN. They can also be deleted with the DEL key or command. Notice that unnamed Direct Data Download files and unnamed Upload/Download files are both stored as "DATA." . They cannot exist in OMEM at the same time with the same name. Any file sent to OMEM will overwrite any existing file with the same name, even if they do not come from the same source.

Since OMEM files do not have to be computed (they already exist in direct waveform datapoint form) they can be rapidly recalled for sequential waveform switching. This is particularly useful in a remote control test environment and is further described in the Remote Programming Section of this manual.

## DIRECTIVE STATEMENTS

Directive statements are used to control and direct the horizontal dimensions of your waveform.

### "FOR" Directive and the variable "T" vs the variable "t":

#### "FOR" Directive

The "FOR" directive is of the form `FOR {time} {expression}`; and it means evaluate the expression for the time duration. Let's use the FOR directive in a few typical statements. `FOR 2.2m SIN(1000 *T)` will generate 2.2 cycles of a 1 Kilohertz waveform (assuming CYC mode) since the specified evaluation time of 2.2 milliseconds (note that seconds are implied) yields 2.2 periods at the specified frequency. `FOR 5 1` produces a one volt level for 5 seconds. Note that the second, again, is implied and that the expression in this case is simply the number 1 which is separated from the time argument (5 seconds) by a space. FOR directives can be concatenated; that is, the multiple statement `FOR .25m 1 FOR 500u COS(1K*t) FOR .25m -1` will produce an output waveform that is plus one volt for 0.25 milliseconds followed by one half cycle (0.5 millisecond) at one kilohertz with a peak amplitude of one volt (times 1 is assumed as the coefficient for the trig term) followed by minus one volt for .25 milliseconds. The use of small or large T to express the variable (time) will be discussed below. It is imperative that a space be left in front of each successive FOR statement to delineate it from the previous one as shown in the foregoing example. Try this multiple FOR statement for drill. The result should be as shown in Figure E-1.

#### "T" Variable Vs "t" Variable:

2045 expressions can contain either no variables or the variable time. The variable time has two notation methods (T or t). Large "T" is the cumulative time from the beginning of the expression (starts coincidentally with the sync out pulse) to the end of the expression (start of the next sync pulse or the end of the waveform depending on trigger mode). In this sense, large "T" can be thought of as "global time".

Small "t" is also cumulative but it resets to zero with each new directive and adds to large "T". Small "t" can be thought of as "local time". Each time that a new FOR statement is encountered small "t" begins at zero but large "T" is continuous and non-resettable. There are further considerations to the relation of "T" and "t" that will be elaborated upon in the REPEAT (RPT) directive discussion which follows in this section.

Figures E-1 & E-2 illustrate the effect obtained relative to the use of large "T" or small "t" with both expressions being otherwise identical. In figure E-1, we used small "t" so that when the second FOR directive was encountered the first data point was taken as Cosine of zero degrees, or plus one volt, which corresponded to the value of the last data point in the first FOR directive, also one volt. In Figure E-2 we used large "T" so that when the second FOR directive was started "T" was at a value of 0.25 milliseconds (1/4 cycle at 1 Kilohertz) and the first data point in the second FOR section was Cosine 90 degrees, or zero volts.

## TIME EXPRESSIONS

Statements of time duration can be entered as expressions. For example, a polynomial can be `FOR(LOG 3.2K*T)m SIN(1K*t)`. In this case, `[(LOG 3.2K*T)m]` is the number of milliseconds that `SIN 1K*t` will be generated. Except for negative time values, there are no limitations to the content of the time expression other than what already apply to any legal expression for a voltage value. The difference between time and voltage expressions is that time expressions are evaluated only once, so the value remains constant once it is computed.

#### "TO" Directive:

The "TO" statement uses the format `TO {time} {constant}` which causes the constant (d.c. voltage value) to exist from the statement start time up and TO the specified time {time}. The statement start time is the current value of large "T" which is zero if this is the first statement. For example, the statement `F22 = TO 10.2m 3.574` would provide

a D.C. value of 3.574 volts from zero time to 10.2 milliseconds, and since we named it, the expression will be stored under the file name F22. "TO" statements can be concatenated. For example, "TO 1 1 TO 2 2 TO 3 3 TO 4 4" would produce a three step voltage staircase composed of one second steps wherein the first level (step) is 1 volt, the second 2 volts, etc. Note that as in the case of FOR statements a space is required to separate the time from the voltage value and to delineate between successive TO statements.

#### "AT" Directive:

The format for the "AT" directive is AT {time} {voltage}. This is similar to the "TO" statement except that a voltage ramp is formed such that the specified {voltage} is achieved at the terminal end, specified {time} of the linear ramp whose starting voltage value is whatever existed at the beginning of the "AT" statement. For example, "TO 1m 0 AT 2m 3 AT 4m -1" would produce a waveform that is zero volts from zero time to one milliseconds which then ramps to plus 3 volts at T = 2 milliseconds at which time it reverses slope and ramps to reach minus one volt at T = 4 milliseconds. You should think of "TO" as a voltage step whereas "AT" is a voltage ramp. From this example, it is obvious that "AT" statements can also be strung together and that they can be combined with "TO" statements. Actually, "FOR" statements can also be included.

#### NOTE:

When creating a segment with a "TO" or "AT" directive, keep in mind that the end time of a TO or AT segment cannot be equal to or less than the end time of the previous segment. So make sure that you allow for one data point of difference. If a segment ends at T = 3.5mS and you issue a TO or AT 3.5mS directive for a new segment, the two segments will overlap and the second segment will have a duration of zero, an impossible condition. . Start the segment with AT or TO 3.51m or whatever is necessary, depending on current time and clock rate, to make room for another data point.

#### MIXED DIRECTIVES ("AT", "TO", "FOR"):

Figure E-3 shows a complex waveform that is constructed with four "AT", five "TO" and three "FOR" statements. This construction should be self evident if the 2045 expression is compared to the waveform for each directive taken separately.

#### REPEAT ("RPT") Directive:

The Repeat [RPT] directive can be used to loop on arbitrary sequences of FOR, AT, and TO segments which we will refer to as "FAT" sequences in the following definitions.

To specify a repeated sequence n times, enter:

.... RPT n (FAT sequence) ....

An example of such a single repeat statement is shown in Figure E-16 where one cycle of a 1 KHz sine wave is repeated twice. If the RPT encloses the entire equation, as in this example, then the 2045 will repeat the entire equation n times, and stop (the output will remain at the voltage of the last point in the equation). The 2045 can, in this manner, generate single or multiple shot arbitrary waveforms by entering RPT n (FAT sequence).

A single level of nested repeats may be used to generate multiple shot waveforms which contain repeated sequences by entering:

RPT n ( . . . RPT n ( FAT sequence ) . . . )

An example of this is shown in Figure E-17 where the inner repeat loop generates two (2) cycles of a 1 KHz Cosine wave; this inner loop is preceded and followed by a ramp. So, the outer loop repeats two (2) times the combination of the first ramp, the two cycles of Cosine wave, and a ramp to zero at 4mS.

## POLY MODE TRIGGERING:

The basic 2045 triggering modes are described in the Standard Function Mode (SFM) of this manual. For POLY mode operation, you must set the trigger mode in a different manner and the trigger mode nomenclature is changed as follows:

### SFM MODE                      POLYNOMIAL MODE

+ START	AT + TRIG (or AT TRIG)
- START	AT - TRIG
+ GATED	FOR + TRIG (or FOR TRIG)
- GATED	FOR - TRIG
+ STOP	TO + TRIG (or TO TRIG)
- STOP	TO - TRIG
+ TO +	AT + TRIG TO + TRIG (or AT TRIG TO TRIG)
- TO -	AT - TRIG TO - TRIG

As in the SFM, "FREE RUN" is assumed if no trigger mode is specified in the POLY mode. Trigger modes can be set in the POLY mode only by prefixing the waveform expression with one of the trigger statements shown above. You cannot set them in the OP 4 configuration as was done in the SFM. When a trigger condition has been properly specified the "MODIFIED" mode will report that trigger condition (in SFM terminology), and the trigger condition (mode) can be changed in the MODIFIED menu, as was the case in the SFM.

For example: FOR TRIG FOR 1m SIN(1K\*T)

Will produce a 1 KHz sine wave, at 1 volt peak amplitude as long as the trigger input is active high (gated).

## REPEAT RESTRICTIONS:

There are four "ground rules" that should be understood when using the Repeat (RPT) directive:

1. Sequences may be repeated a minimum of 1 and a maximum of 65535 times. Repeats may be strung sequentially in a single equation.
2. There is no minimum number of points required in each repeated sequence.
3. The generator will automatically choose the Clock Rate but this can be overridden by a user specified Clock Rate (refer to POLY mode Modifiers in this Section).
4. An outer level RPT (i.e. a RPT applying to the entire equation) puts the 2045 into a single shot mode, such that on completion of the waveform the output stops. This is the case with figures E-16 and E-17, which will not run continuously. To view these waveforms, you need a scope with storage capabilities. However, without the outer RPT the 2045 will cycle the waveform continuously. The effect on the various triggering modes with and without the outer loop RPT function can be seen in the following descriptions.

**TRIGGER CONSIDERATIONS WHEN USING REPEAT:****FREE RUN**

Equation output will start when the computer has finished computing. If there are no outer level RPTs, it will run indefinitely, stopping only when the R/S key is toggled. If there is an outer level RPT, the waveform will run for the specified RPT number and then stop.

**AT  $\pm$  TRIG or  $\pm$  START**

Signal output will start at the next active trigger edge and will run the specified RPT number of times until complete. The next trigger edge will restart the output. Trigger edges occurring while generating an equation will be ignored (the 2045 is not retriggerable). There is a minimum dead time of 0.5  $\mu$ s from equation end until a trigger edge is accepted. If there is no outer RPT the waveform will run continuously after being triggered.

**TO  $\pm$  TRIG or  $\pm$  STOP**

Signal output will start when the computer has finished computing. The signal will stop after it has either run the specified RPT number of times to completion, or an active trigger edge has been detected. Subsequent triggers will have no effect because the output has already stopped. To restart the function output, RUN/STOP must be toggled. With no outer RPT, the waveform will cycle continuously until trigger.

**AT  $\pm$  TRIG, TO  $\pm$  TRIG, or  $\pm$  TO  $\pm$** 

Signal output will begin at an active trigger edge, and will stop after it has either run the specified RPT number of times to completion, or until a trigger edge has been received. The next trigger edge will restart the signal. With no outer RPT, the signal will alternately begin and stop at each trigger.

**FOR  $\pm$  TRIG, or  $\pm$  GATE**

In the gated mode, the output signal will remain active while the trigger input is active or until the equation has run the specified RPT number of times to completion. The output will stop when the trigger goes inactive, and will restart when it goes active again. With no outer RPT, the signal will begin whenever trigger is active and stop when inactive.

**DATA BLOCK LENGTH RESTRICTIONS:**

As mentioned earlier, waveform points are generated in blocks of 64. If the number of points called for by the equation is not a multiple of 64, the last point computed will be repeated up to to the next block boundary. These repeated "fill" points can produce a visible discontinuity between cycles of wave shapes such as a continuously running sine wave.

If the discontinuity is undesirable, one possible way to fix this is to follow a FOR statement with a second FOR statement that is preceded by a RPT (repeat directive). The idea of the repeats is to call for enough continuous cycles to produce a total number of points that is a multiple of 64. The actual number of repeats to specify must be calculated. The need to follow the first FOR with a repeated FOR is because an outer level RPT will produce only a single shot waveform.

Keep in mind that the preceding is only a suggestion on how you might deal with potential discontinuities when calling for a continuous, repetitive function in POLY mode. If all you want is a continuous sine wave, it is better to use Standard Function mode where discontinuities will not occur.

## POLY MODE (ARBITRARY) WAVEFORM EXAMPLES:

The best way to become familiar with the power and flexibility of the 2045 when using the "POLY" mode is to try some examples. In doing this, you will find that the syntax is intuitive and the machine is essentially self teaching. Single "FOR" statement examples (Figures E-4 thru E-13):

Figure E-4 shows an AM modulation example in which a 5000 hertz sine wave is being modulated by a 500 hertz sine wave. Note that since both terms are expressed with unity amplitude coefficients, the modulation index is 100 % and the peak envelope amplitude is 2 volts. To vary the modulation index the low frequency term would be multiplied by a coefficient of other than one and you would have to include an outer parenthesis pair around the multiplier and the sine term. Don't forget that implied multiplication is not done on the 2045, that is, you need the "x" operator which is displayed as an asterisk. Try changing the modulation index on this example.

Figure E-5 is similar to the previous (Figure E-4) example except that the times "X" operator was changed to an addition "+" operator so that the result is a summed 5000 hertz with 500 hertz, rather than a modulated, multiplied, composite.

Figures E-6, E-7, and E-8, show FM modulation expressions using linear, logarithmic, and exponential sweep functions, respectively.

Figure E-9 is a linear sweep similar to that shown in Figure E-6 except that a square wave carrier is used in this case.

Figure E-10 shows a linear function of time ( $1500 \times t$ ) multiplied by an exponential function of time ( $e$  raised to the minus  $3000 \times t$ ).

Figure E-11 shows a 1.5 Kilohertz carrier modulated by an exponentially increasing ramp function.

In Figure E-12, we asked for a 500 hertz sine wave for one half cycle (1 millisecond) and since the trigger mode is FREE RUN, the function repeats continuously to produce a Full Wave Rectified pattern, as shown.

Figure E-13 shows a  $(\sin X)/X$  pattern. Note that the peak of the function was offset from zero time by the  $(t-a)$  notation.

## MODIFIERS (S/N, OFST, MARK, CLK, FILT):

Modifiers are quite different from the previously described DIRECTIVES. Whereas a "directive" such as FOR, TO, AT, etc. is an integral part of the waveform expression; a "modifier" such as S + N, OFSET, etc. is appended to the end of a waveform expression as a special instruction which can be changed at RUN TIME by invoking the "MODIFIED" mode.

### S + N (Signal plus Noise):

When the flashing CURSOR is one or more locations past the last character in a waveform expression, depressing the "S/N" key will cause two value fields to appear. Namely, [NAMP =] and [NBW =] which stands for Noise Amplitude and Noise Bandwidth, respectively. The operator is then expected to enter the noise parameters (Amplitude and Bandwidth) to produce the desired "signal plus noise" condition. For the allowable values relative to these noise parameters refer to the NOISE field description in the Standard Function mode section.

To obtain a specific Signal-to-Noise ratio, the operator must specify a particular noise amplitude (in rms volts) and noise bandwidth based on the operators' knowledge of the rms signal value and signal bandwidth that is represented by the waveform expression. Figures E-14 and E-15 illustrate the use of the NOISE MODIFIER. In Figure E-14, we have synthesized a pulse modulated R.F. carrier signal that resembles a LORAN-C navigation signature. Since an actual receiver output would generally be corrupted by noise, this real life situation has been emulated in Figure E-15 by specifying a noise MODIFIER in the form of  $NAMP = .025$  (units are understood to be in volts) and  $NBW = 200\text{KHz}$ . Because of the restrictions on the noise bandwidth, the closest available legal value of NBW is selected (See item 4, Noise Mode, in section D). This value may be viewed in MODIFIED mode. The example given in Figure E-15 has the noise 'frozen' to the signal. For a more realistic waveform, repeat the signal many times so that each cycle has different noise samples. The number of TARGET POINTS (Set in OP 4) can be increased to match the increase in cycles.

#### OFST (Offset):

A D.C. offset term can be specified independently of any offset that may be implied by the mathematical waveform expression. If the evaluated waveform expression contains a d.c. term, the externally specified d.c. term (by means of OFST) is simply added to the mathematically implied offset. The peak signal plus offset cannot exceed the output full scale range. This offset value is specified at the end of the waveform expression by keying OFST followed by the value in volts with an optional equals sign. For example:  $OFST = .3$  or  $OFST .3$  or  $OFST 300\text{m}$  would all result in a positive 0.3 volt offset; whereas  $OFST -.3$  would yield a negative 300 millivolt offset. A complete typical equation might look like this:

$FOR\ 1\text{m}\ PI * SIN(1K * T)\ OFST\ .3$

This example would produce a one Kiloherzt sine wave with a peak amplitude of 3.14159 volts (PI) with a plus 0.3 volt d.c. offset. Note that the evaluation period, 1 millisecond, is one period at 1 Kiloherzt.

#### MARK (Marker Output):

A rear panel "Marker" output is provided to produce a positive logic level transition at a time {marker time} after the leading edge of the "Sync Pulse" (also a rear panel signal). Like all other MODIFIERS this is specified at the end of the waveform description. For example:

$FOR\ 1\text{m}\ PI * SIN(1K * T)\ OFST\ .3\ MARK\ 156\text{u}$

This is the previous example with a MARKER specified at 156 microseconds. As in the case of all MODIFIERS, an equal sign is optional, meaning we could have specified  $MARK = 156\text{u}$ . Since the marker time is only available in multiples of 32 clock periods the actual time delay will be the closest obtainable.

#### FILT (Filter):

One of two internally available Low Pass Filters can be applied to the Output Function by specifying  $[FILT = ]$  followed by the filter cut-off frequency which can be either 2 MHz or 20 MHz. Then to apply a 2 MHz filter to the waveform of the previous example we would have:

$FOR\ 1\text{m}\ PI * SIN(1K * T)\ OFST\ .3\ MARK\ 156\text{u}\ FILT = 2\text{M}$

Again, the equal sign is optional.

**CLK (Clock):**

This refers to the waveform evaluation interval used by the 2045 to construct the Output Function. If you use a CLK: modifier, it should be placed after the equation. If you don't specify a clock rate with the CLK: modifier, the 2045 will divide the waveform duration by the number of target points to determine the closest available clock rate, then calculate the actual number of points to be computed to the nearest block of 64. If a particular Clock Rate is forced by the CLK modifier that requires more output memory than that which was specified in the [OP 4] TARGET PTS configuration, the specified clock rate will dominate and the number of solution points will automatically be extended as required to achieve the waveform duration time at the specified evaluation (clock) interval; provided of course that sufficient memory is available and the clock rate is achievable. Appropriate ERROR messages will warn the operator if either of the foregoing limitations are encountered as the result of a requested clock rate and number of points. The 2045 will not compute a waveform under these error conditions. The clock modifier (CLK) is appended at the end of a waveform description as in the previous examples.

Try out the following polynomial:

FOR 1m SIN(40K\*t)/(8-(e^(2K\*t)))

The above would be evaluated at a clock rate of 1 us that results from a TARGET PTS set-up of 1000 points (ref OP 4) and a FOR statement of 1 millisecond. In this example, the discrete data steps (1 us each) are readily apparent. If better horizontal resolution is desired, you can force a particular clock rate as a "modifier" at the end of the waveform expression. To achieve this, we can specify CLK = 40n to yield a resolution every 40 nanoseconds instead of the original 1000 nanoseconds (1 us). Note that TARGET PTS in the OP4 menu is still 1000. However, forcing the particular clock rate of 40 nanoseconds now requires a 25000 point solution since the FOR statement (in the waveform description) still requires a one millisecond execution time. Since something must give, the TARGET PTS 1000 point statement is automatically overridden.

It should be noted here that the same 40 nanosecond clock resolution could have been achieved without the CLK MODIFIER by restating the TARGET PTS setup configuration as 25000 points in which case the default clock would have been 40 nS ( $1 \text{ ms}/25000 = 40 \text{ nS}$ ). If we had tried to increase the clock rate (decrease clock period) by going to MODIFIED mode and changing the CLK = 1u to CLK = 40n the horizontal resolution would not have changed. What would have changed is the waveform execution time which would have gone from 1 millisecond to 40 nanoseconds since the clock statement in the modified mode refers to the output clock and not the evaluation clock. The evaluation clock determines horizontal resolution while the output clock determines output frequency.

**IMPORTANT:** All modifiers must be separated from the body of the POLY expression by at least one space, and modifiers can be strung with space separators, for example:

{POLY expression} [space] CLK = 1u [space] FILT = 20K etc.

## GENERATING SWEPT WAVEFORMS:

The 2045 can synthesize waveforms using its single independent variable time. The general form of sine functions generated on the 2045 is  $V = \text{SIN}(\Phi(t))$  where  $\Phi(t)$  represents the phase angle of the sine wave and has units of cycles or radians depending on the calculation mode selected (CYC/RAD). If  $\Phi(t) = f*t$ , then  $f$  has units of cycles/second (hertz) or radians/sec. The instantaneous frequency of this function can be calculated by evaluating the derivative of  $\Phi(t)$  at time  $t$ .

$$\text{FREQ}(T) = d \Phi(t) / dt$$



A 5 ms sweep from 1 KHz with a sweep rate of 2 KHz/1 ms can be described on the 2045 as :

FOR 5m SIN(INT(1K+2K/1m\*t)) CLK = 1u  
 or:  
 FOR 5m SIN(1K\*t+2K/1m/2\*(t ^ (2))) CLK = 1u

Note: INT refers to the Integral (INT) key on the 2045.

#### Exponential Sweep Example:

To generate an exponentially swept sine wave with base frequency B and a sweep rate of 1 decade/m sec gives  $FREQ(t) = B * (10 ^ (t/m))$ . To produce 2 decades of an exponentially swept sine wave with a base frequency of 1 KHz and a sweep rate of 1 decade per 2.5 ms you could use:

FOR 5m SIN(INT(1K\*(10 ^ (t/2.5m)))) CLK = 10n

However, if you enter and compute the equation with the clock rate as shown above, you will notice the display says:

"Computing . . . 500000 points  
 Press RESTORE to suspend computation."

After about 15 seconds, if you press RESTORE, the display will show:

"Computation is 0% complete  
 Would you like to continue? (Y or N)"

If you press the YES key, the computation will continue. But you might wonder why the computation has not yet reached even 1%. This is because computing 500000 points for this particular equation will take the better part of an hour. A 10nS clock rate with a complex equation such as this one requires a rather lengthy computation interval. So unless you need that many points, you should press the NO key which will display the top line of the equation and "Computation has been aborted" on the second line. Press the right arrow key and the abort message will disappear, leaving the equation. Hold the right arrow key to move the cursor to the CLK modifier and change the value to 1u. Press ENTER. The display will indicate it is computing 5000 points. After 15 seconds, pressing RESTORE will indicate that computation is very nearly done. Considering that the number of points is 99% fewer than before, this is not surprising. Now if you press YES the waveform will shortly be computed and available at the output.

An interesting experiment would be to sweep out the filters on the 2045 with an exponentially swept sine wave:

FOR 20u SIN(INT(100K\*(10 ^ (t/10u)))) CLK = 10n FILT = 2M

This will show the swept frequency response of the 2045 internal 2 MHz analog filter.

Note: The 2045 can only compute trigonometric functions with arguments up to 170K cycles (1M radian). Waveforms requiring more than 170K cycles may be split into 2 or more waveform segments.

#### MATH TOPICS:

**MATH RANGE:** Math operations are performed over the numerical range of 1.7 EE-38 to 1.7 EE +38 with 24 bit resolution.

**TRIGONOMETRIC RANGE:** The maximum argument for trigonometric functions is 170K cycles (1M radians).

**EXPRESSION RANGE:** A waveform expression, including modifiers and file name, can contain up to 1280 displayable characters.

**PARENTHESES NESTING:** Parentheses nesting levels are unlimited.

#### **MAGNITUDE NOTATION:**

[EE] key used for Engineering Powers of Ten Notation

Example [EE] [3] indicates times 1000

[2nd] [M] key sequence indicates Meg (X 1000000)

[2nd] [K] key sequence indicates Kilo (X 1000)

[2nd] [m] key sequence indicates milli (X 1/1000)

[2nd] [u] key sequence indicates micro (X 1/ 1000000)

[2nd] [n] key sequence indicates nano ((X 1/ 1000000000))

#### **SPECIAL CONSTANTS:**

[2nd] [e] key sequence indicates the constant 2.718 . . . . .

[2nd] [pi] key sequence indicates the constant 3.141 . . . . .

### **MATH FUNCTIONS:**

The following Math functions all operate in the same format:

**FOR...AT...TO** [time] [Math function](expression)

Every Math function will calculate its appropriate operation based on whatever immediately follows it enclosed in parentheses. The parentheses are important. You will notice that pressing the key corresponding to any of the following functions will bring up on the display the abbreviation of the function name followed by an opening parenthesis. Whatever value or expression you choose to follow the function with must be enclosed in parentheses, so all you have to do is add a closing parenthesis at the end of the expression.

1. SIN will calculate the sine value of the expression.
2. ARCSIN will calculate the Arc Sine value of the expression.
3. COS will calculate the Cosine value of the expression.
4. ARCCOS will calculate the Arc Cosine of the expression.
5. TAN will calculate the Tangent of the expression.
6. ARCTAN will calculate the Arc Tangent of the expression.
7. Y<sup>x</sup> key (exponentiation) is used to raise an expression [Y] to a power [x]. For example, to raise 1000 times T to the negative third power the key sequence would be: [1] [2nd] [K] [x] [T] [Y<sup>x</sup>] [-] [3] [)] Note that the right parenthesis is used to close the left that was automatically given just after the Y<sup>x</sup> key is depressed, so that in this example when the display is viewed there will be a left parenthesis before the minus sign (system inserted) and a right parenthesis after the 3 that you inserted.

8. INT denotes Definite Integral function when used in the format:

[FOR] [x] [INT] [exp]

where:

[x] is a time duration over which integration is done

[exp] is the expression to be integrated.

9. LOG will calculate the common logarithm (base 10) of the expression.
10. LN will calculate the natural logarithm (base e) of the expression.
11. ABS will evaluate the expression according to its Absolute Value.
12. SGN will determine the Signum (Sign) value of the expression, and consider it as one of three values: -1 if negative, 0 if equal to zero, or +1 if positive. This is useful for generating pulse and square wave type functions.

### Polynomial Evaluation Order:

It is very important that the operator understand the order in which his various equation terms will be evaluated by the 2045. This evaluation order is:

*First:* All Parentheses pairs from innermost to outermost.

*Second:* Functions; for example, SIN, LOG, etc. NOTE that (y to the x) is NOT a function. If there is more than one function they are evaluated, in order, from Left to Right.

*Third:* Multiplication, Division, and Exponentials (y to the x). In the case of multiple operations the order is Left to Right.

*Fourth:* Addition and Subtraction with the order again being Left to Right.

**IMPORTANT:** Since Exponentials are not given a higher priority than Multiplication and Division you may get solutions that you would not have normally expected with reference to the generally accepted algebraic evaluation ranking. To overcome this difficulty, it is good practice to include an outer parentheses pair around the expression which is to be raised to a power and another pair around the expression that represents the power. This insures that the y raised to x operation is given a higher priority than multiplication and division.

### Negative Time:

Another idiosyncrasy of 2045 math that may lead you astray is the fact that negative T cannot be used. This can be overcome by notation fix in which you multiply T by minus one. That is instead of -T use -1\*T and instead of -t use -1\*t. Of course, the numeric value could be other than one; for example, -500\*T etc.

## MORE WAVEFORM EXAMPLES:

Figures E-18 through E-23 further demonstrate some of the 2045's capabilities.

Figure E-18 shows a simple sine wave at 1MHz with a CLK modifier specified at 1.25n. Assuming the default TARGET PTS value of 1000 points is in effect, the CLK modifier in this case proves useful in eliminating discontinuities by forcing the computation of more points. To illustrate this, remove the CLK modifier and look at the result.

Figure E-19 illustrates the use of the Signum (SGN) function. The same sine function used in figure E-18 has SGN added to it, which produces a square wave at the same 1MHz frequency.

Figure E-20 also uses the same sine function of figure E-18 with the ABS (Absolute value) function applied to it. The result is the same as full wave rectification.

Figure E-21 shows a linearly swept sine wave.

Figure E-22 demonstrates the Signum function when applied to the swept sine wave of figure E-20.

Figure E-23 makes use of the RAD (Radians per second) mode with two FOR directives and ARCSIN functions.

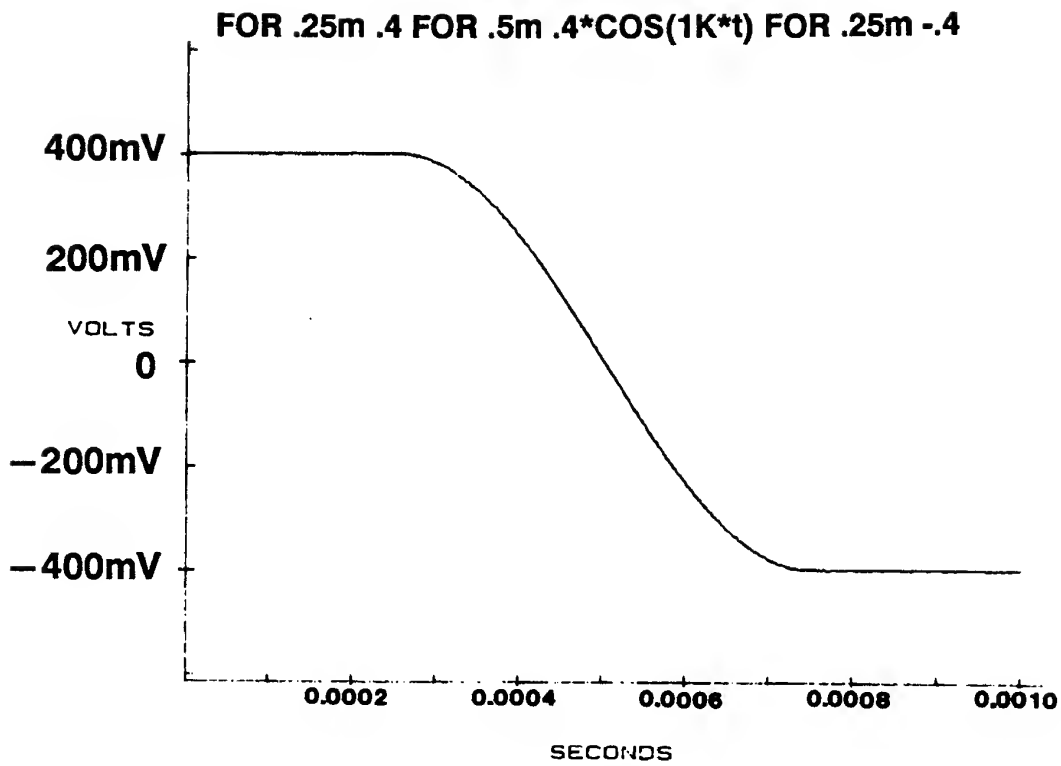


Figure E-1

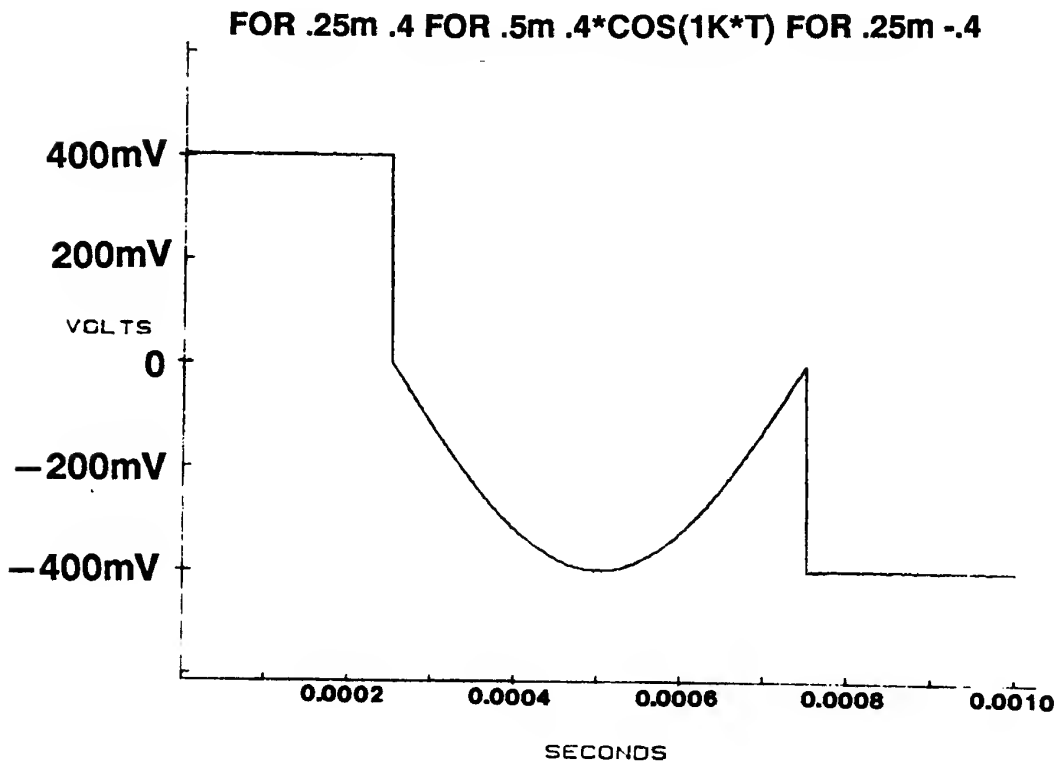
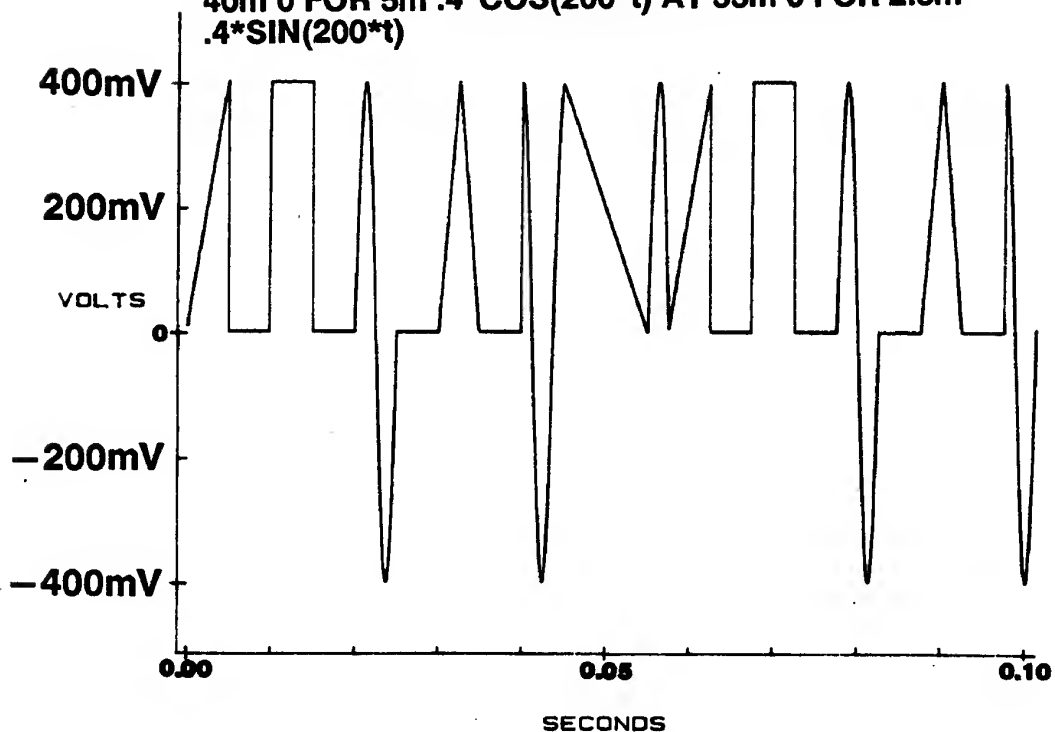


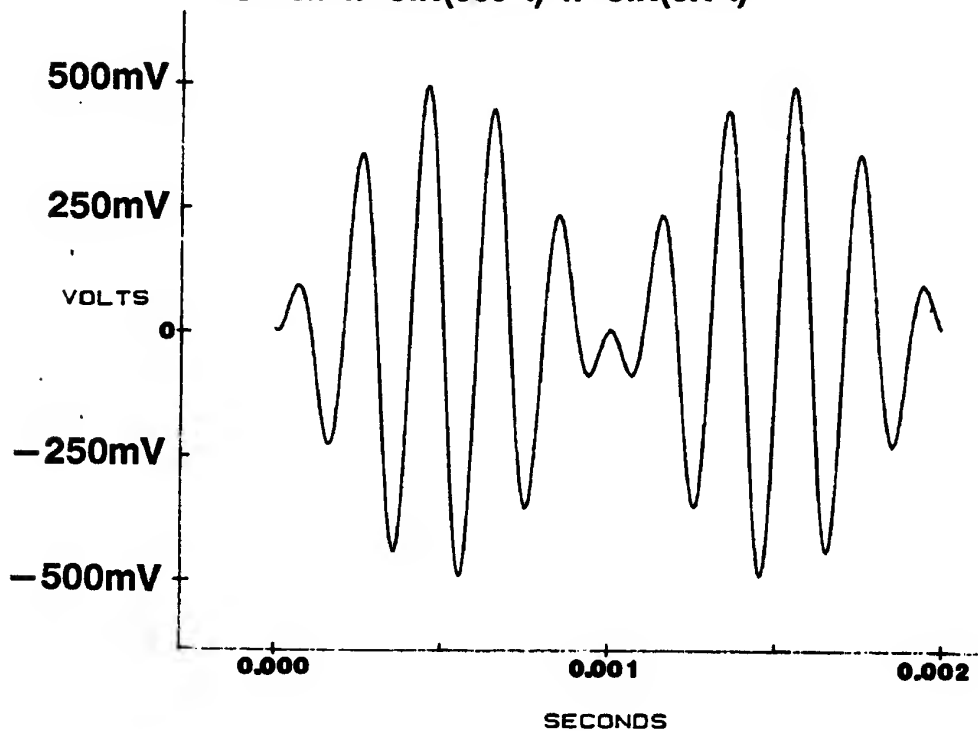
Figure E-2

**AT 5m .4 TO 10m 0 TO 15m .4 TO 20m 0 FOR 5m  
.4\*SIN(200\*t) TO 30m 0 AT 32.5m .4 AT 35m 0 TO  
40m 0 FOR 5m .4\*COS(200\*t) AT 55m 0 FOR 2.5m  
.4\*SIN(200\*t)**



### Figure E-3

**FOR 2m .7\*SIN(500\*t)\*.7\*SIN(5K\*t)**



### Figure E-4

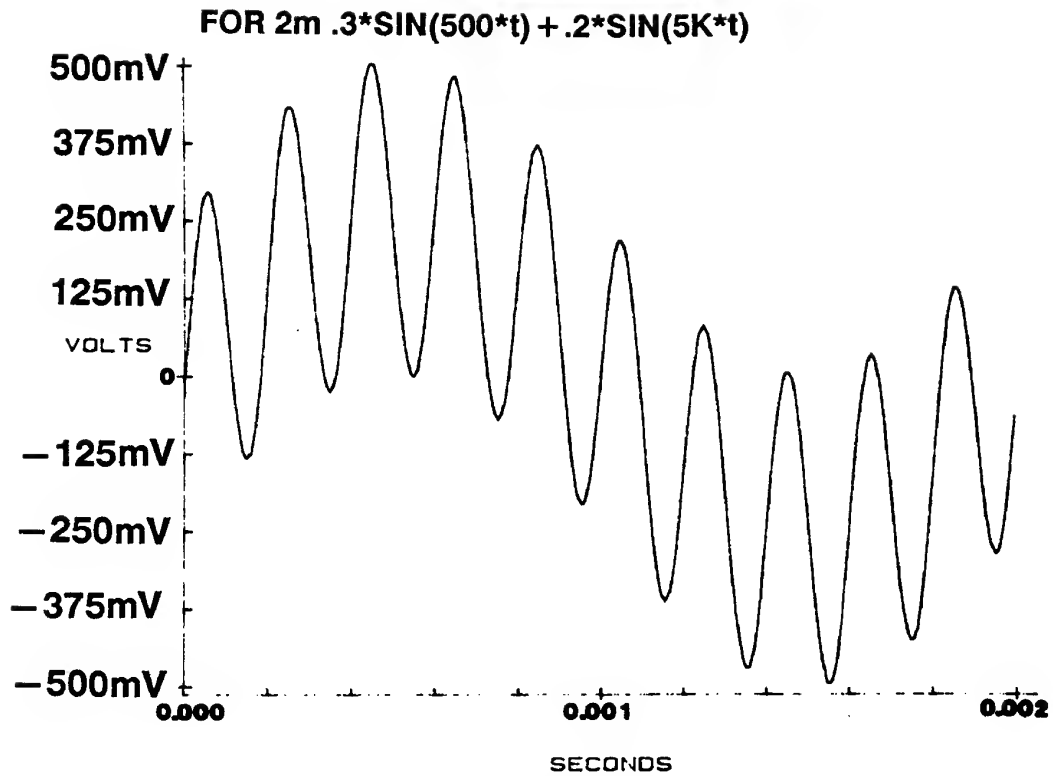


Figure E-5

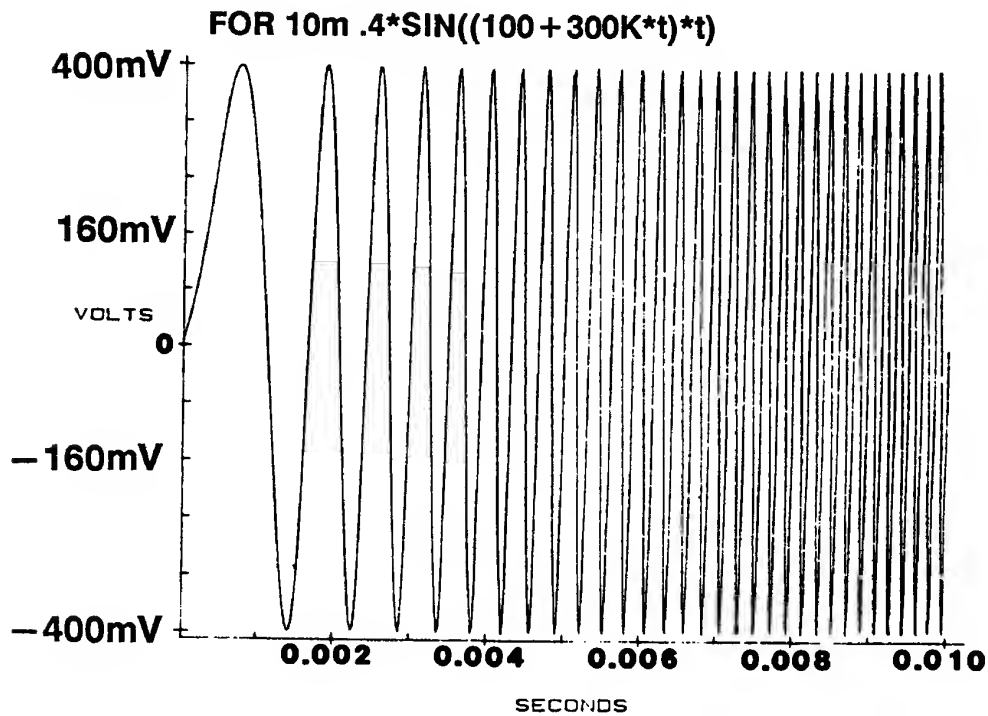


Figure E-6

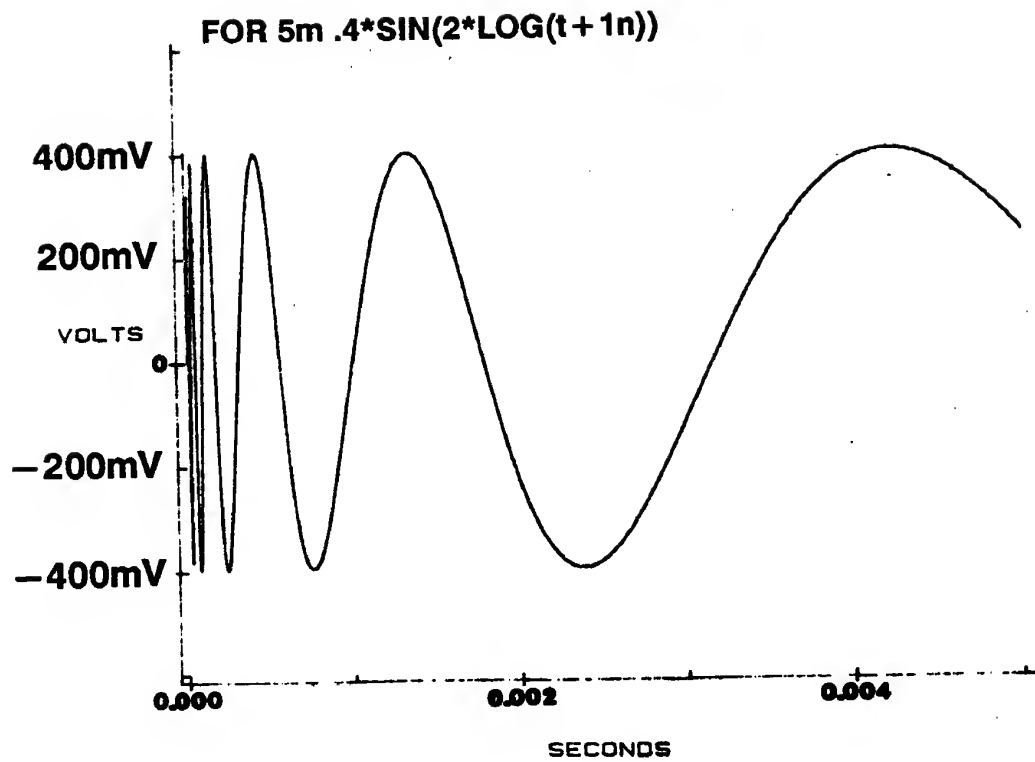


Figure E-7

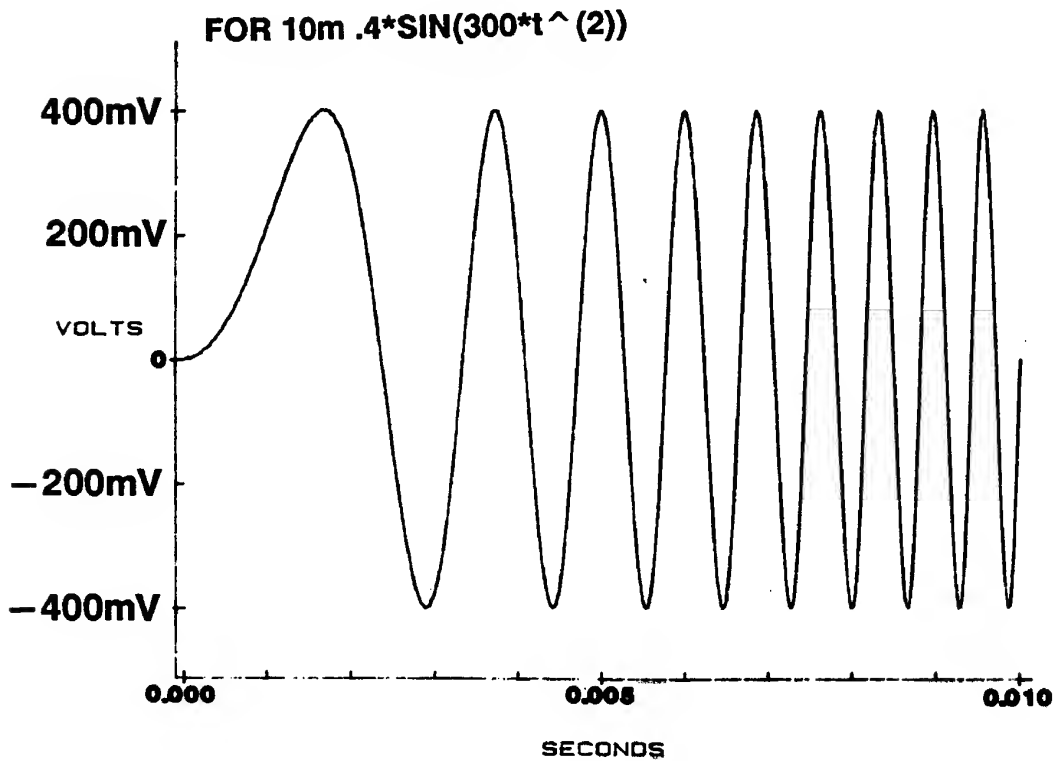


Figure E-8



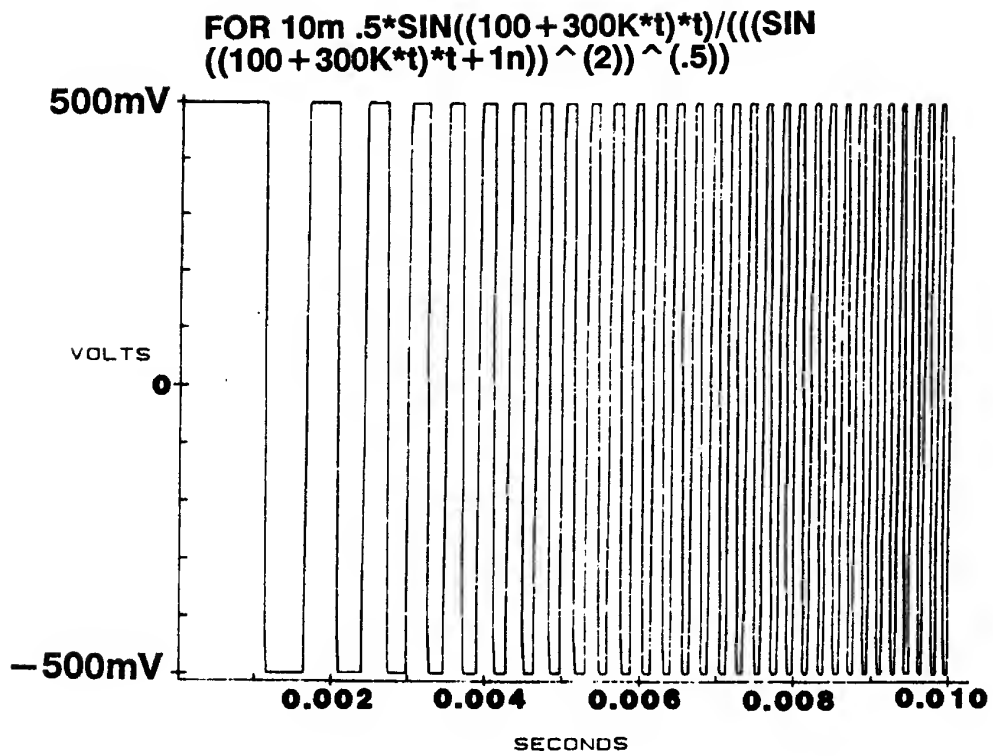


Figure E-9

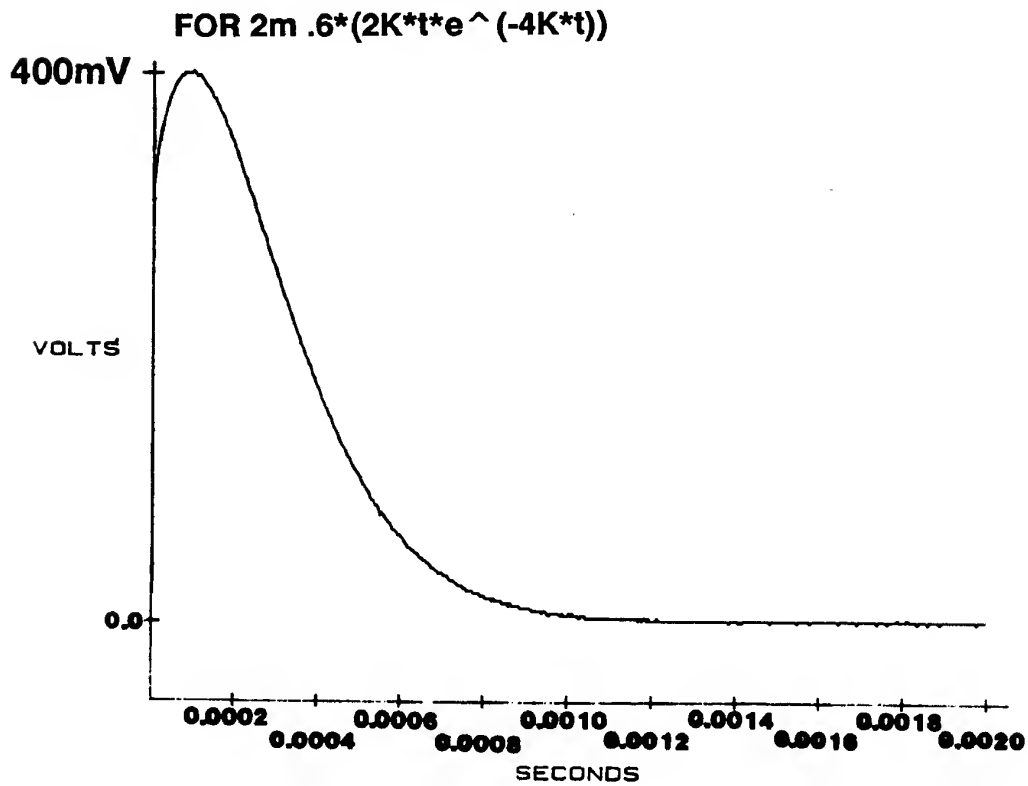


Figure E-10

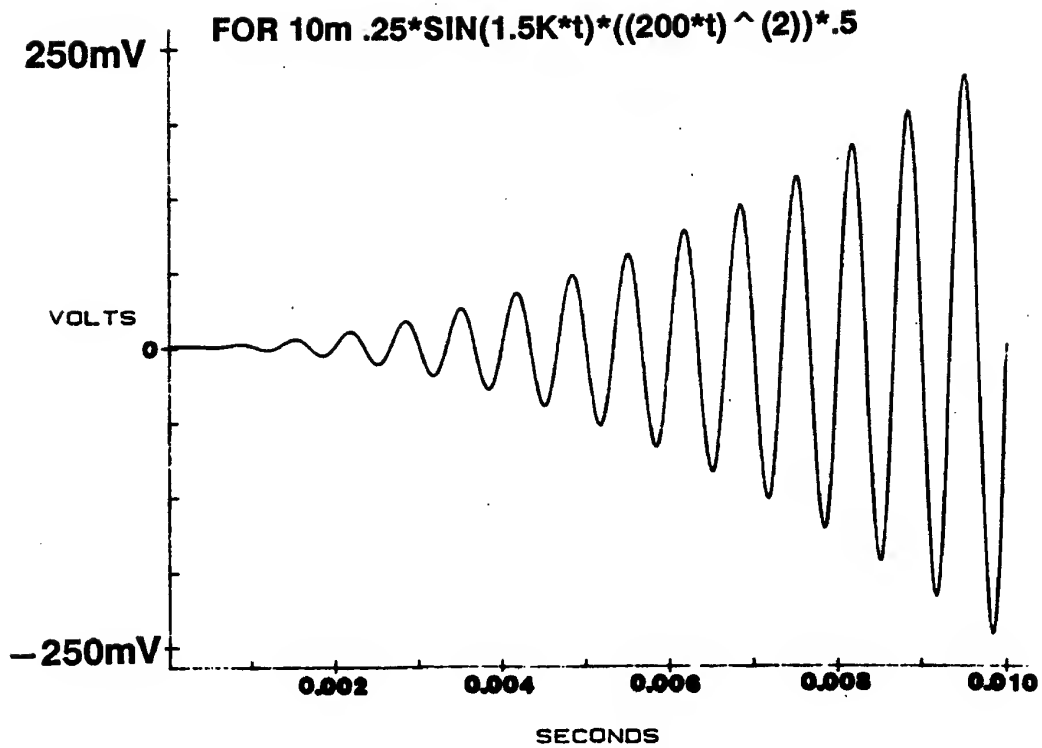


Figure E-11

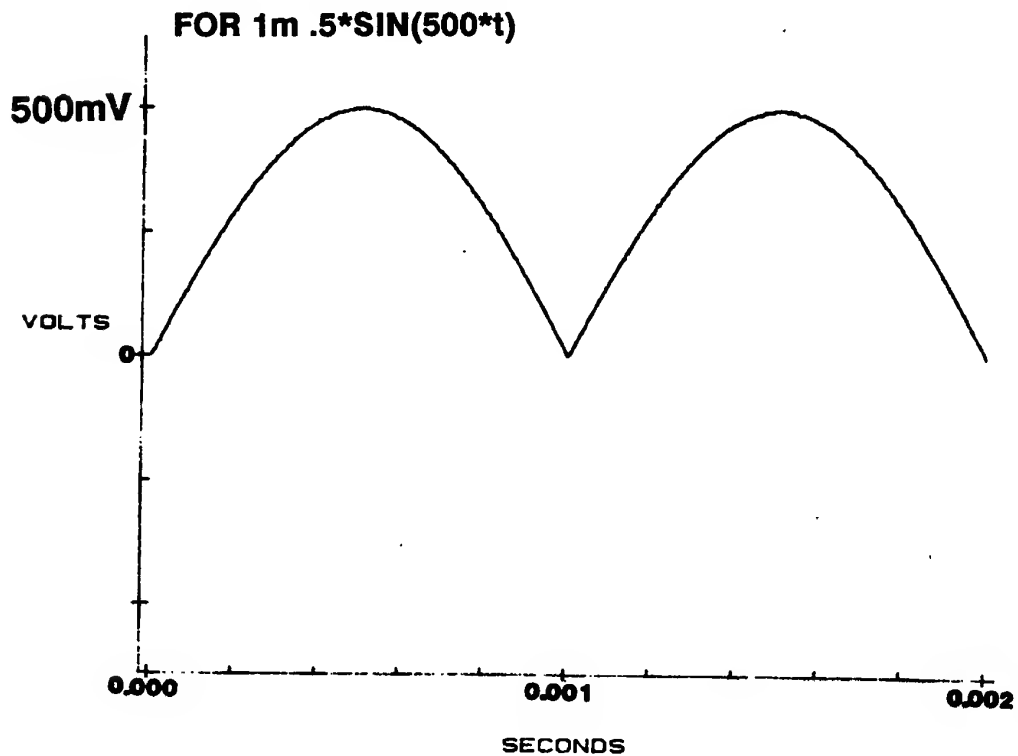


Figure E-12

FOR 20m .077\*SIN(1+1K\*(t-10m))/(1+1K\*(t-10m))

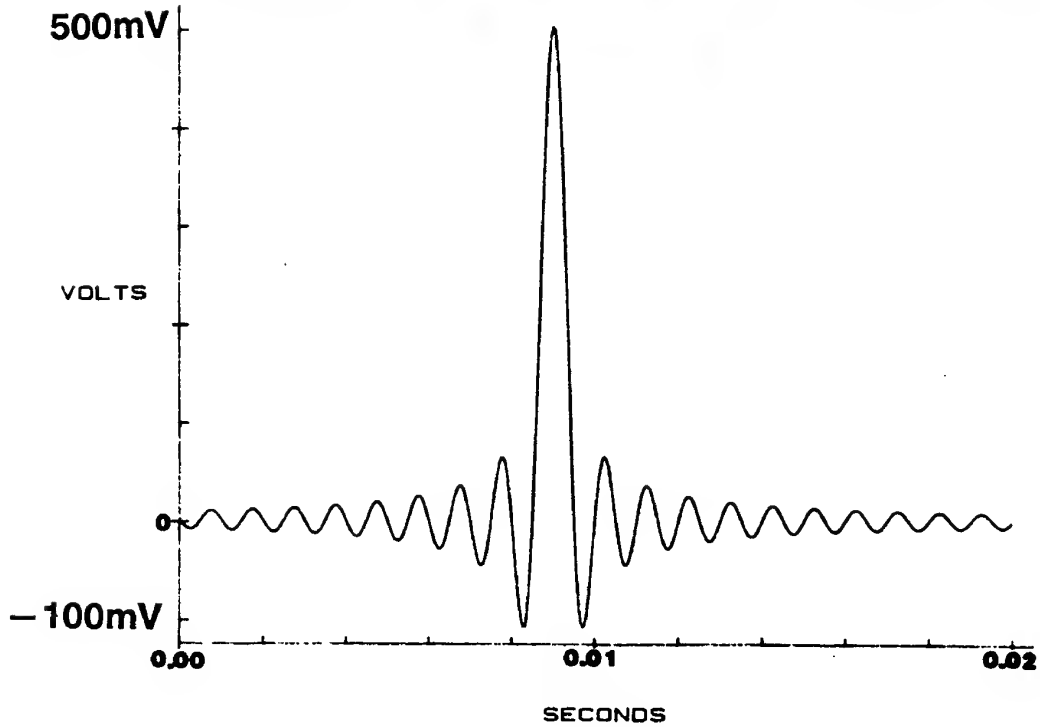


Figure E-13

FOR .4m (16K\*t^(2))\*(e^(-30.7K\*t))\*SIN(100K\*t)

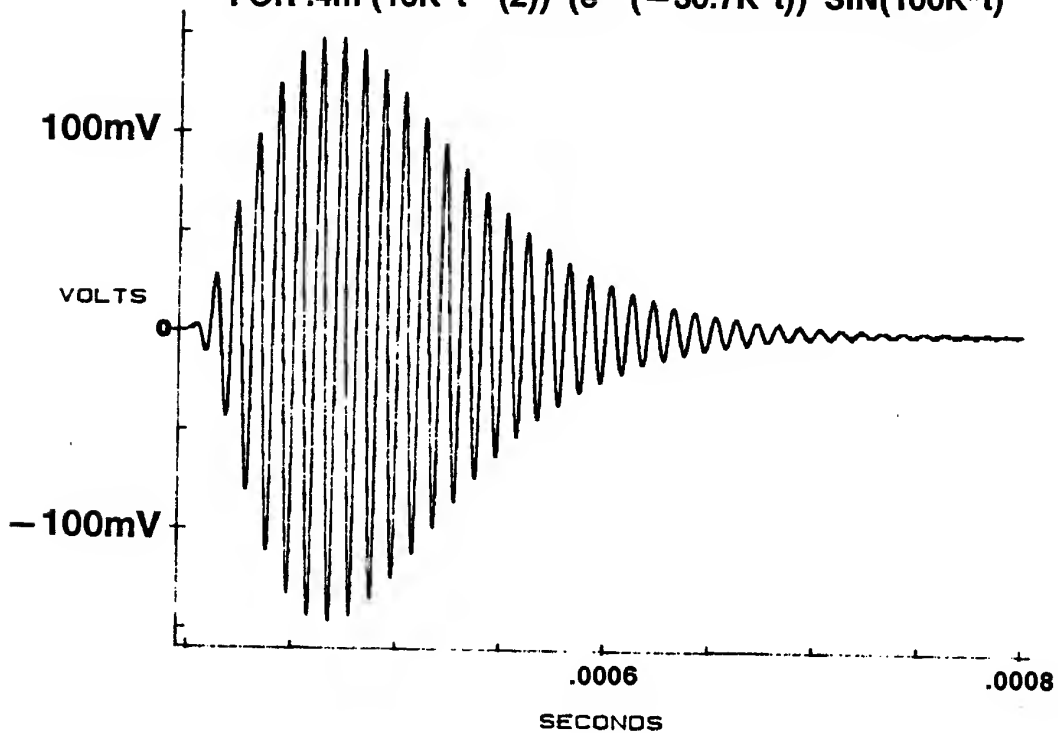


Figure E-14

FOR .4m (16K\*t^(2))\*(e^(-30.7K\*t))\*SIN(100K\*t)  
 NAMP .015 NBW 200K

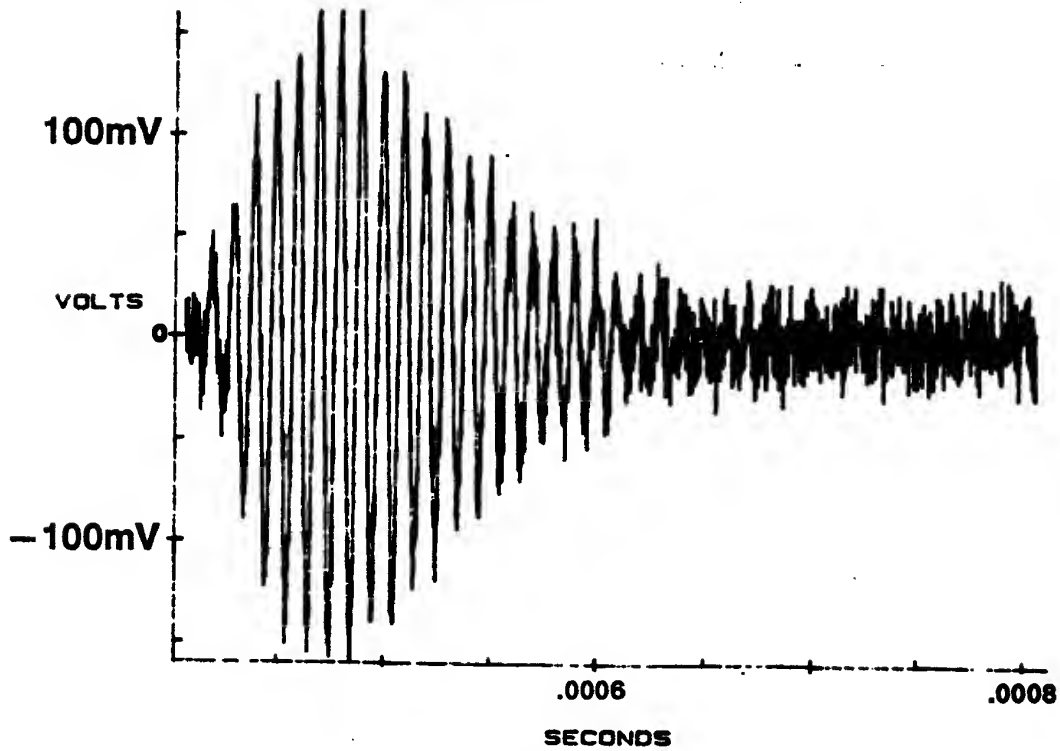


Figure E-15

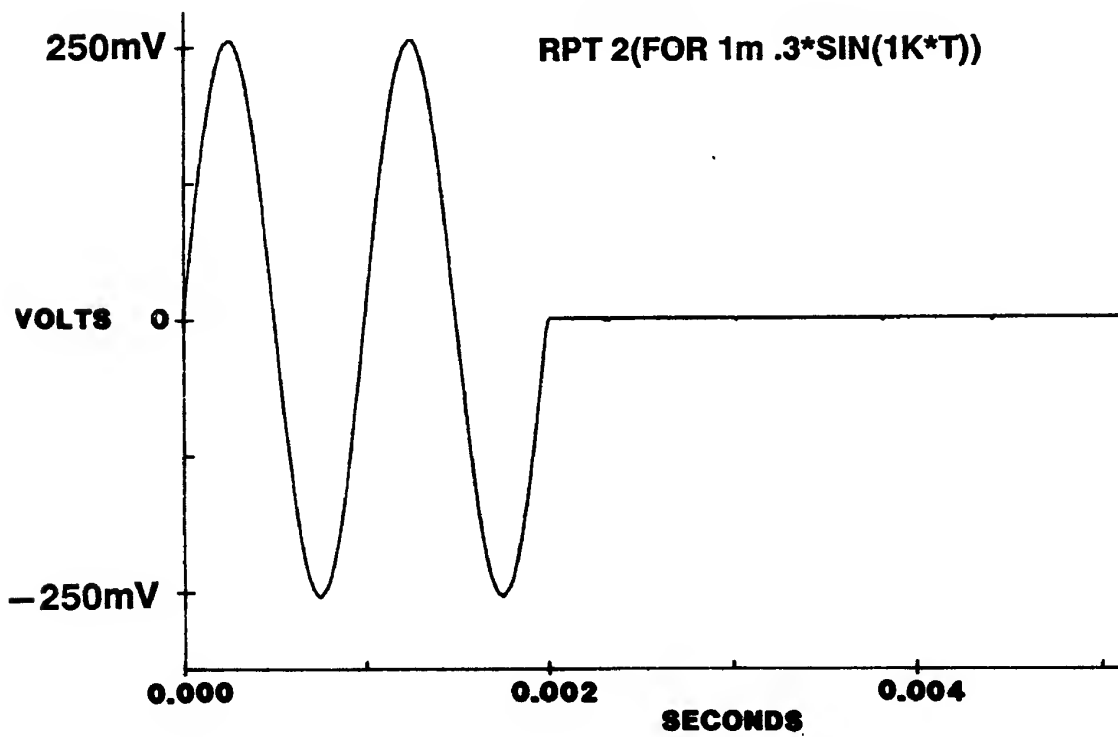


Figure E-16

2045

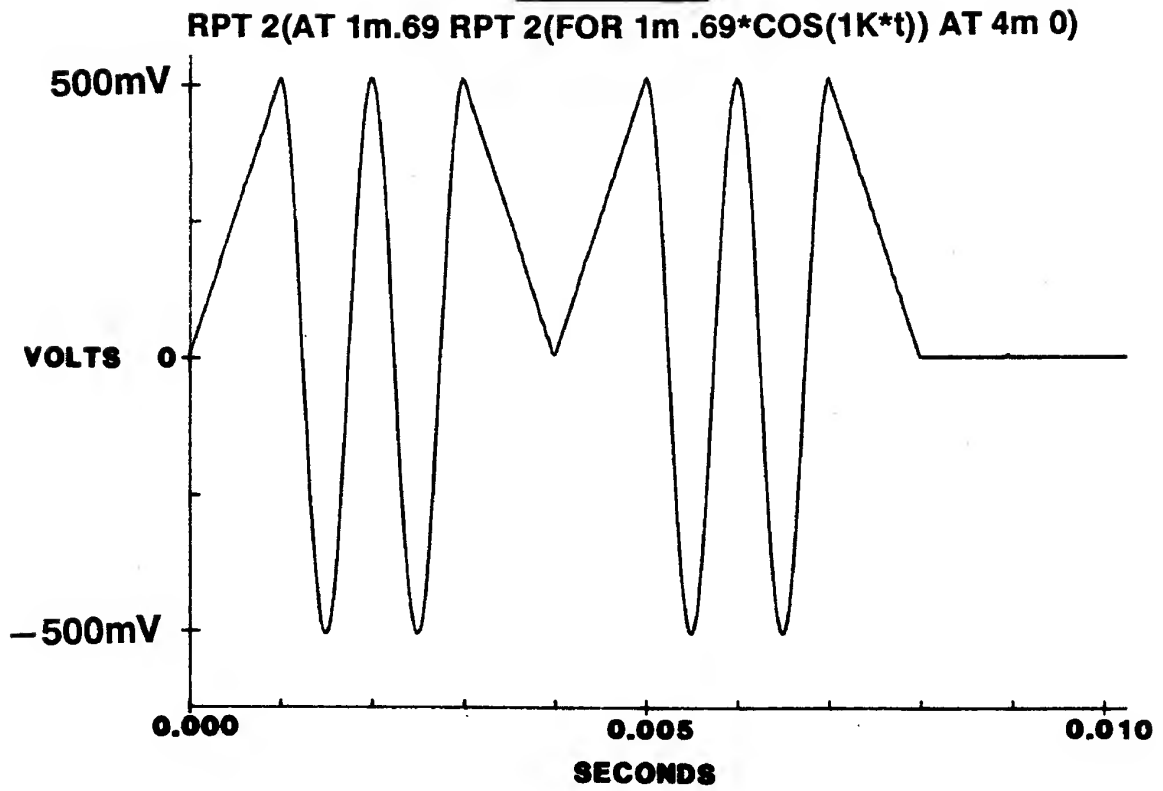


Figure E-17

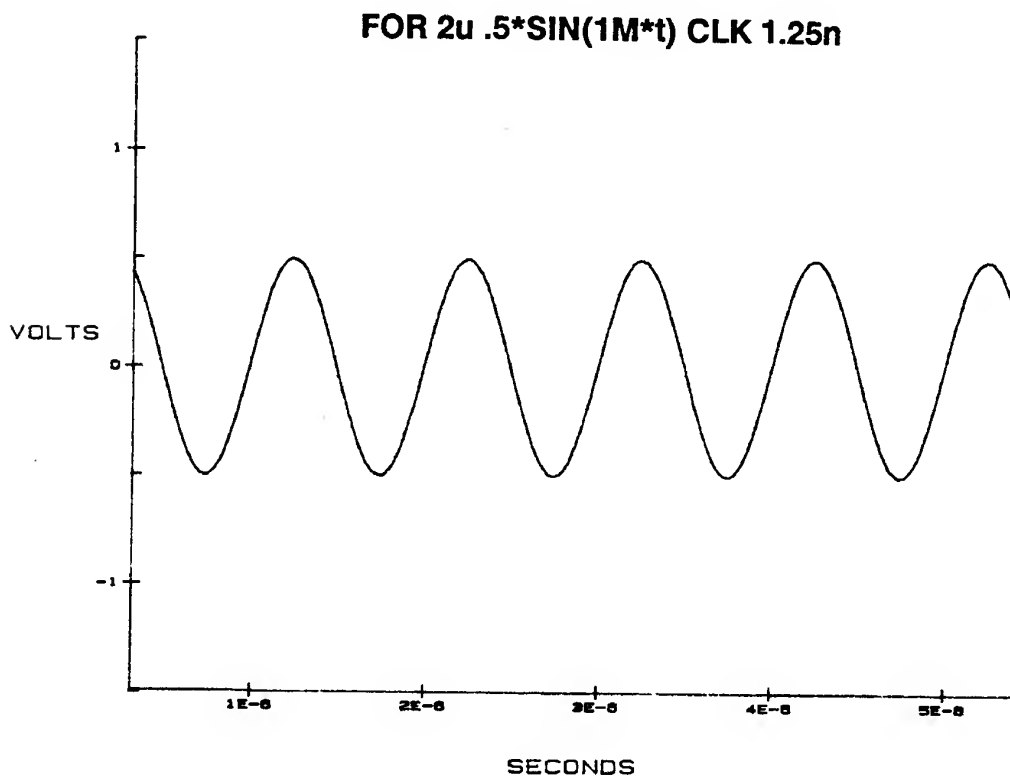


Figure E-18

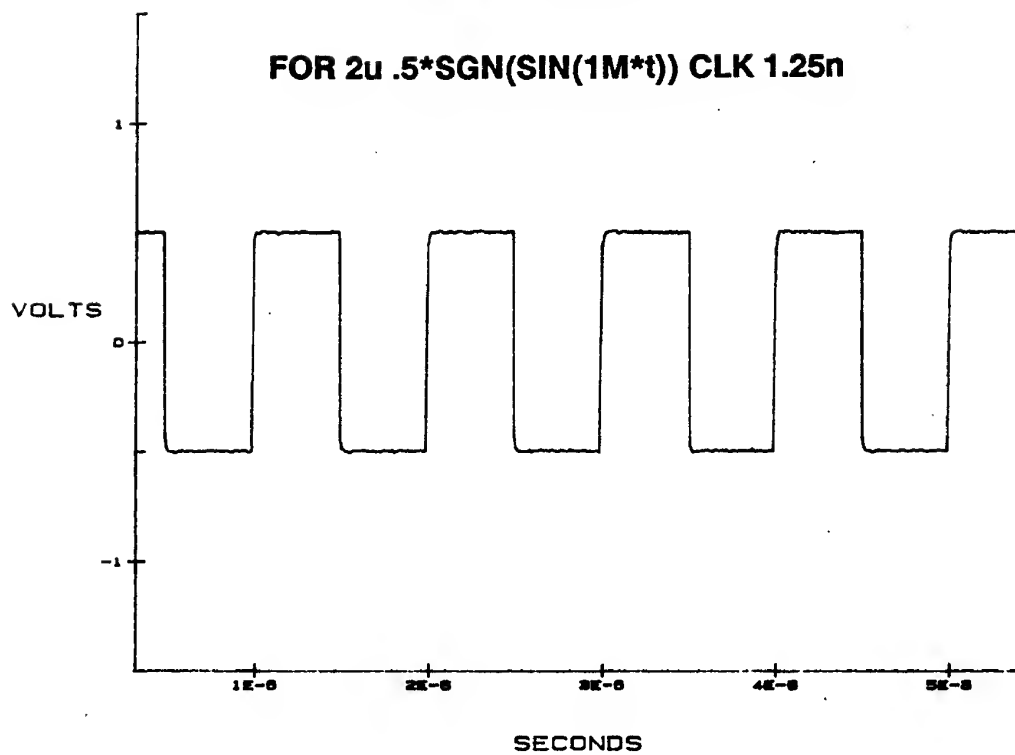


Figure E-19

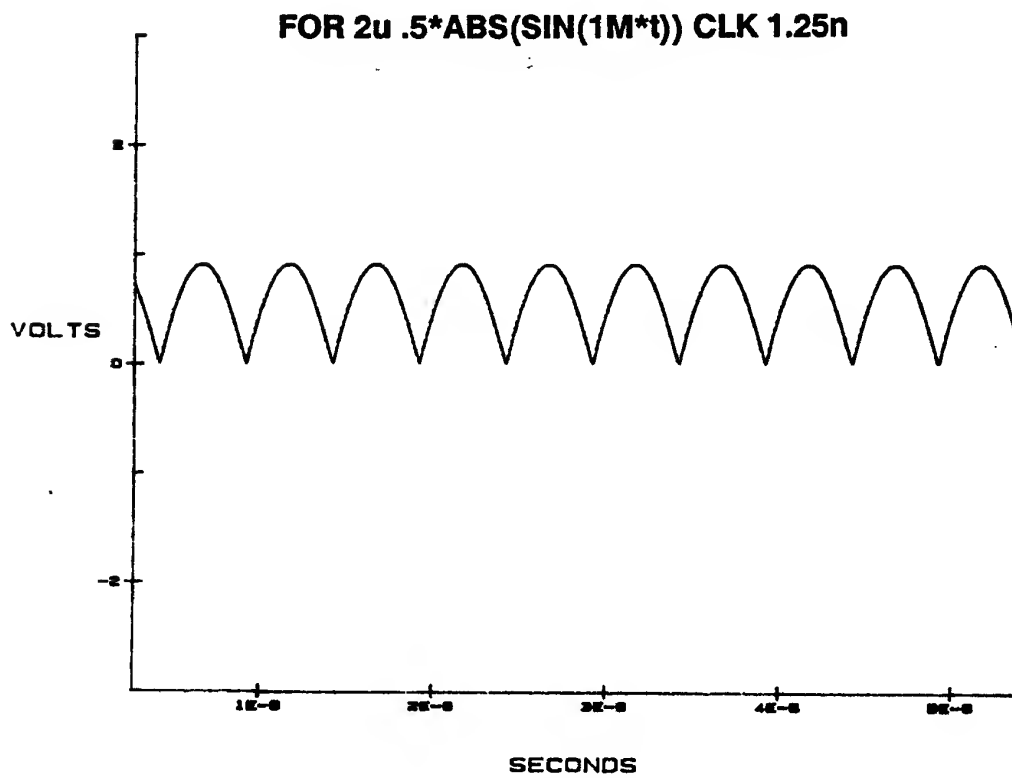


Figure E-20

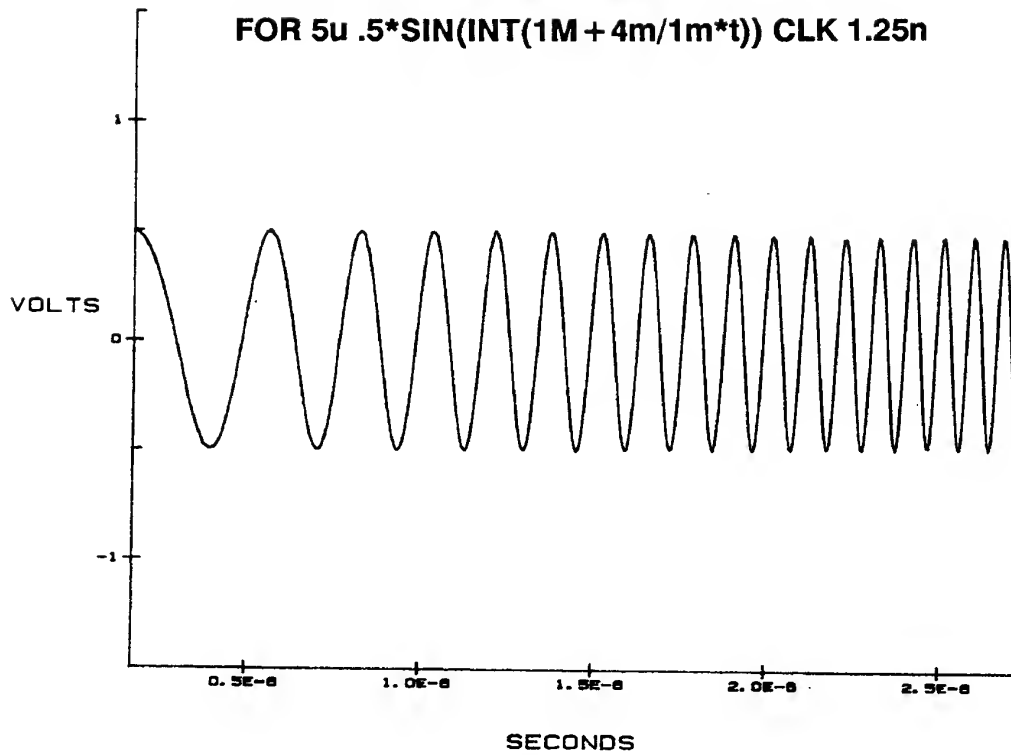


Figure E-21

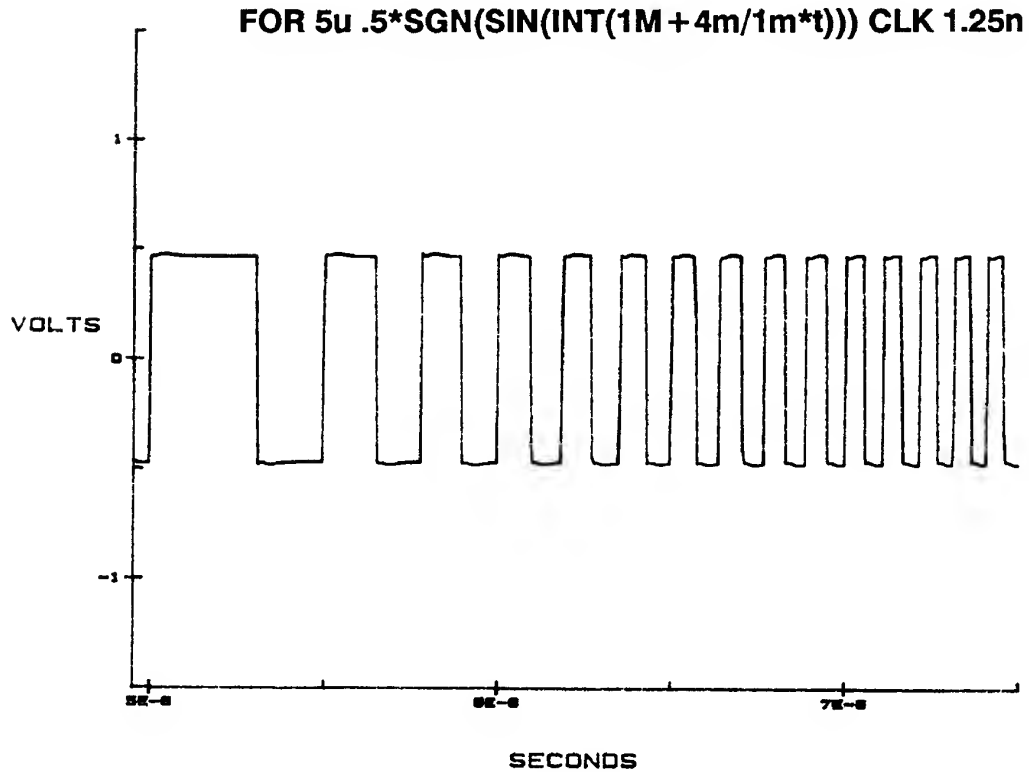


Figure E-22

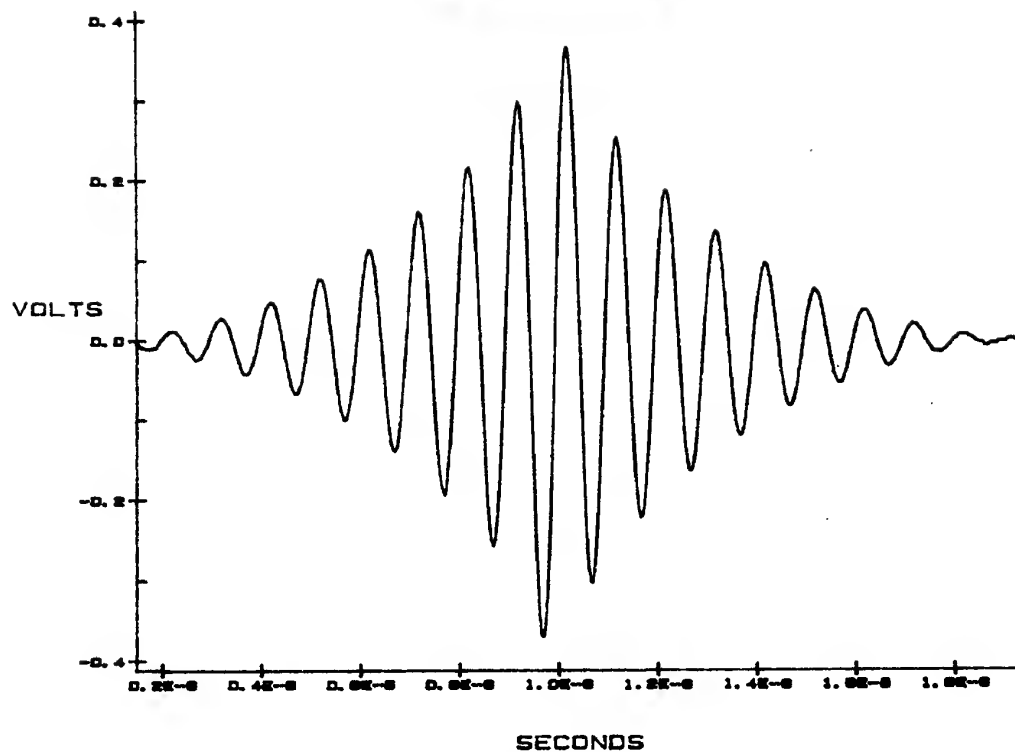


Figure E-23

```

FOR 1u .3*ARCSIN(1E12*(t^(2)))*SIN(2*PI*
10m*t) FOR 1u .3*ARCSIN(1E12*((1u-t)^(2)
))*SIN(2*PI*10m*t) CLK 1.25n      [RAD Mode]

```





## MODIFIED MODE

This very useful feature is designed to monitor waveform parameters and to make temporary hardware adjustments so you can modify waveform output without having to recompute. Since the computed data remains unaffected by anything you change in MODIFIED mode, this lets you experiment easily with instantaneous results of any adjustment.

The MODIFIED operating mode can be entered directly from either the Standard Function or Polynomial (arbitrary) modes by pressing the RESTORE key or sending the remote command MOD. When MODIFIED mode is invoked, the MODIFIED LED will light and nine menu fields will appear on the display (six at a time).

The menu appears as shown:

CLK = 10n	S + N = 0.00000	OFST = 0.00000
FILT = NONE	MARK: 0.00000	TRIGM = FREE RUN
NOISE = ON	NAMP = 0.00000	NBW = 250.640K

While in MODIFIED mode, the output function is active (assuming R/S is in the RUN state) and unchanged unless one or more of the field values or attributes are modified. With a couple of exceptions, field values or attributes are changed in the same manner as in Standard Function mode. You can key in a value with the number keys or slew the value with the arrow keys. An attribute list can be stepped through with the arrow keys. With either a value or an attribute, the modification is executed by depressing the ENTER key, whereupon the modification can be seen at the actual output if the R/S (RUN/STOP) mode is RUN. To exit MODIFIED mode, press the RESTORE key. This will cause the MODIFIED LED to go out and all field parameters that were temporarily changed in MODIFIED mode will revert to their original values in whatever mode you entered from.

All of these fields will contain the current value and/or attribute for the indicated menu parameters. They will be familiar because you set them in the Standard Function mode menu fields, except for Clock (CLK = ) and Signal plus Noise (S + N).

### CLOCK (CLK = )

The value in the clock field reports to you the clock period (in seconds) that was used to synthesize the current output function. When you invoke a change to the CLOCK period in MODIFIED mode, you are changing the output clock period but not the clock period that was used to derive the function. Therefore, you will observe a FREQUENCY change of the output and not a change in horizontal resolution with the frequency fixed, as you might intuitively expect. The clock rate (period) that the 2045 uses to derive the function is obtained by dividing the waveform period by the number of solution (waveform) points requested. If this quotient is less than the minimum clock period available then the minimum available clock period is used and the number of solution (waveform) points is correspondingly reduced.

**SIGNAL PLUS NOISE (S + N = )**

The S + N field represents the peak to peak total possible excursion of the waveform.

Where:

S + N value = (2\* AMP) + (10\* NAMP)  
 AMP = signal peak voltage  
 NAMP = RMS noise voltage

Because all MODIFIED mode adjustments are strictly temporary hardware modifications, changing the value of S + N allows the output amplitude to be varied up or down without requiring recomputation of waveform data points. The signal to noise ratio remains constant with any S + N modifications.

**MARKER (MARK:)**

MARK is given in seconds. It has an alternate format, MPTS, which is given in points. This can be changed back and forth with the FORMAT key.

OFFSET (OFST =)  
 FILTER (FILT =)  
 TRIGGER MODE (TRIGM =)

These three MODIFIED menu fields all operate in the same way as in Standard Function mode.

NOISE (NOISE =)  
 NOISE AMPLITUDE (NAMP =)  
 NOISE BANDWIDTH (NBW =)

Of the three noise related fields, only the first one, NOISE, can be modified to either ON or HOLD. NAMP and NBW are read only fields in MODIFIED mode and cannot be selected for change. Pressing the FIELD key when the cursor is under NOISE will advance the cursor back to CLK.

## REMOTE PROGRAMMING AND CONTROL

Remote programming and control is accomplished with either the RS-232 or GPIB (IEEE-488) communication options. Interface details are described in Section B of this manual.

Before utilizing the remote programming features of this instrument, we suggest that you gain a thorough understanding of the material presented in all previous sections of this manual. This section is presented on the basis of that prerequisite.

Remote programming can be separated into two basic schemes. In the first, an external program controller is using the instrument to create waveforms in a manner that would correspond to that used by a manual operator; i.e., Standard Functions and Polynomial (arbitrary) mode. We will refer to this as "Remote Control." In the second case, the external controller (host) will be downloading direct data that will be used to generate waveforms and/or digital patterns that are determined by the external host. We will refer to this as "Remote Programming".

### REMOTE CONTROL EXAMPLE:

The following example shows the use of remote control mnemonics to generate a pulse train at a 10 KHz rate with a 1 microsecond pulse width. The pulse amplitude is to be 3.25 volts with a 0.5 volt negative d.c. offset. And, we wish to corrupt the pulse with 0.25 volts rms of noise where the noise bandwidth is to be 20 KHz. In addition, we desire that the SYNC output occur 2 microseconds before the leading edge of the output pulse which corresponds to 98 microseconds from the previous pulse leading edge in the output pulse train.

#### MNEMONIC      COMMENT

STOP    ↵	Disables the Function Output
SSQR    ↵	Selects Standard Function SQUARE WAVE mode
FREQ = 10K    ↵	Sets frequency to 10 KHz
AMP = 3.25    ↵	Sets pulse amplitude to 3.25 volts
PLSW = 1u    ↵	Sets Pulse Width to 1 microsecond
OFST = -.5    ↵	Sets Offset to minus 0.5 volts
DLY = 2u    ↵	Sets Delay to microseconds

Note: in this example we could also have used DLY = -98u

NBW = 20K    ↵	Sets Noise Bandwidth as close as possible to value selected
NOISE = 0    ↵	Activates Noise source
NAMP = .25    ↵	Sets Noise Amplitude to 0.25 volts rms
RUN    ↵	Executes the Output Function

Note that the trigger mode is FREE RUN. The (↵) sign, in the above program indicates "carriage return". The parameter value or control command is ENTERED automatically by the "carriage return" as opposed to the use of the ENTER key when the 2045 is being manually operated.

If while the above function is running we wish to change a parameter, the revised mnemonic would be sent for that parameter only. For instance, if we wish to change the pulse width to 3 microseconds the following mnemonic would be sent:

PLSW = 3u

This example parameter change (Pulse Width) requires recomputation of the waveform which will take on the order of 200 milliseconds. Parameter changes that involve hardware attributes only and do not require recomputation such as Amplitude, Offset and Filter ('A' output only) will take place approximately 35 milliseconds after receipt of the "carriage return".

## REMOTE PROGRAMMING (DIRECT DATA DOWNLOAD):

The 2045 can generate analog waveforms that are externally composed and downloaded via a remote host. Direct Data downloads are done from a host computer using RS-232 or GPIB IEEE-488. The transfer rate is in accordance with the selected BAUD rate if RS-232 is used. If you use GPIB IEEE-488, the transfer rate is approximately 2K Bytes/sec unless Fast I/O (high speed DMA) is selected in OP488; in which case, the transfer rate is approximately 300K Bytes/sec.

## DOWNLOAD FILE TYPES:

There are two types of files, named and unnamed.

1. Unnamed files are preceded by the ASCII string "DATA". Only one unnamed file can exist in output memory. Loading another file, computing a Standard Function, or POLY function will purge the unnamed file. The unnamed file will appear in the file directory as "DATA". The appended dot (.) indicates that this is a binary (output memory) file and it is therefore volatile. Do not send the dot (.) with the file name, it is automatically appended in the 2045.
2. Named files are preceded by the ASCII string "DATA <filename>" where the filename will be an Alphanumeric string of up to 8 characters. For example, if you send "DATA SCHMATA", the data which follows will be logged in the directory under the name "SCHMATA.". Note the appended dot which indicates that it is a volatile output memory binary file (just as with an unnamed file, "DATA."). Up to 16 data files can reside in output memory at one time. However, since there is always one file in output memory, "STDFN." or "POLY." (whichever was last computed), the maximum number of files which can be transferred via direct data download is 15. On the other hand, the maximum size of the file(s) in output memory is 524288 points.

The advantage to this format (which we refer to as segmented output memory) is that any one of these files can be recalled in the conventional manner {RCL FILENAME} and run without having to take the bus time necessary to download the file, or the computation time necessary to calculate a POLY or SFM expression. The access time is approximately 35 milliseconds from the carriage return that follows the RCL command to the time that the new recalled waveform appears at the Output.

## DOWNLOAD FILE CONSTRUCTION:

A data download file consists of three main parts:

1. The file is always preceded by the ASCII string "DATA" or "DATA <filename>" which indicates that a direct data file is to follow.
2. A MAIN HEADER always follows the opening ASCII string (DATA). The Main Header conveys hardware set-up parameters such as; clock rate, amplitude, offset, filter bandwidth, noise amplitude and bandwidth, etc. There can be only one Main Header except in the case of segmented (multiple) output memory files which will be discussed later.
3. A Data Set consisting of a DATA HEADER, waveform DATA, and CLOSING HEADER follows the Main Header.
4. The Data Set (item 3 above) can be repeated depending upon whether repeat and/or constant compression sections are contained in the waveform.

### MAIN HEADER:

The main header contains the hardware settings appropriate to the waveform being downloaded. It is 32 bytes long and is coded in the following manner:

Parameter	# of bytes	Starting byte	Remarks	See note
Clock Period	4	0	IEEE F.P. units in Seconds	1,2
Offset	4	4	IEEE F.P. units in Volts	1,2
Signal Amp.(pk-pk)	4	8	IEEE F.P. units in Volts	1,2
Noise Amp. (rms)	4	12	IEEE F.P. units in Volts	1,2
Noise Bandwidth	4	16	IEEE F.P. units in Hertz	1,2
Filter Cut-off	4	20	IEEE F.P. units in Hertz	1,2,3
Trigger mode	2	24	Unsigned 16 bit integer	2,4
Not Used	2	26	0, 0 (only)	5
Not Used	4	28	0, 0, 0, 0 (only)	5

You will note that only waveform parameters are passed in the Main Header. Any attribute that is not associated with the Output connectors is directly controlled with the appropriate mnemonic as given in section H. For example; if you wish to invoke a Marker Output at 83.5 microseconds; send "MARK=83.5u" at any convenient point in the program.

If you wish to change a Main Header parameter without having to re-send the entire program, send the mnemonic MOD to enter MODIFIED mode and then send the new parameter in accordance with the mnemonic listing given in section H. To revert to the original parameter value, send RESTORE.

**NOTES:**

1. IEEE Floating Point format is a 4-byte number representation. For computer languages like "C" and "Pascal" there are simple subroutine calls to get the IEEE floating point number. For "BASIC" language, we have written a simple conversion routine which is given in this Section entitled IE3FP.BAS. Additionally, as a quick reference, we have listed some common conversions from base 10 to IEEE in Tables G-1, G-2 and G-3.

2. The IEEE byte values are sent as the ASCII number representing the value of the byte (0 thru 255). In the case of Hewlett Packard computers using BASIC, CHR\$ is used to make the conversion to ASCII numbers. On an IBM compatible PC using BASIC, use STR\$.

3. There are two internal L.P. filters, 2 MHz and 20 MHz, in addition to NONE which yields the full output response bandwidth. If you specify a FILT value (locally or remotely in a polynomial, remotely in a standard function) other than 2MHz or 20MHz, the 2045 will automatically choose one of the available filters according to certain thresholds as follows:

<u>Entered Value</u>	<u>Resulting Filter</u>
FILT = 0	NONE
FILT = 1	2M
FILT = 6.3M	2M
FILT = 6.4M	20M
FILT = 31.6M	20M
FILT = 31.7M	NONE

4. The numerical specification for Trigger Mode is as follows:

0	FREE RUN
1	+ START
2	- START
3	+ GATED
4	- GATED
5	+ STOP
6	- STOP
7	+ TO +
8	- TO -

5. The last block of 6 unused bytes is required in the download sequence and is to be filled with zeros. This space is reserved for future expansion capabilities.

**DATA HEADER:**

The header follows the main header and it tells the 2045 what type of data is to follow and how it is to be executed.

This header is 10 bytes long and is presented in the following format:

<u>Parameter</u>	<u># of bytes</u>	<u>Starting byte</u>	<u>See note</u>
Segment Type	2	0	note 1
Segment Length	4	2	note 2
Constant Value	4	6	note 2, note 4

**DATA HEADER NOTES:**

1. Segment Type: The choices for Segment Type with their two byte descriptors are as follows:

0, 1 Block type - used when there is a single set of data to be outputted with no repeated segments.

0, 2 Constant - used when a constant value is to be outputted for a period of time and the programmer does not want to simply fill successive memory locations with the same value.

0, 3 END - denotes that this is a closing data header and determines the end of a download sequence.

0, 4 Reserved - do not use.

0, 5 Repeat - denotes that the data set is to be repeated "n" times where "n" is determined by the "segment length" parameter.

0, 6 Repeat End - denotes the end of a repeated segment, see note 3 below on the use of loop counters.

2. Segment Length: The use of this field will vary depending upon the segment type (see note 1.) as described below:

<u>If Segment Type =</u>	<u>The Segment Length is:</u>
0, 1	Unsigned 32 bit integer representing the length of the data block in two 16 bit words.
0, 2	A 32 bit unsigned integer representing the number of clock pulses that determine the duration of the constant (d.c.) portion. Note that the d.c. value is sent in the constant field (bytes 7 & 8).
0, 3	Fill with 0, 0.
0, 5	The first two bytes are the Loop Counter to be used (see Note 3 below) which are either 0,0 or 0,1. The last two bytes are the length of the repeat segment expressed as a 16 bit integer value.
0, 6	The first two bytes are the Loop Counter to be used (see Note 3 below). The last two bytes are unused and must be filled with 0, 0.



3. Loop counters keep track of the number of times that a waveform or waveform segment is to be repeated (looped upon). There are two loop counters 0, 0 and 0, 1. They can both count up to 65535 but are used differently. The programs RPTTEST0.BAS and RPTTEST1.BAS included in this Section illustrate their use. In RPTTEST1.BAS, loop counter 0,0 is used because this is an "n" shot repeat. That is, the entire waveform is repeated N times. The other counter (0, 1) is associated with multiple repeats and may be included inside (nested in) the 0, 0 counter but; 0, 0 may not be included in 0, 1. Be aware that the nested loop counter cannot be allowed to end on the same clock cycle as the outer loop counter when a clock period faster than 15 nanoseconds is being used. For clarity, we can relate these two loop counters to their use in the POLY expression mode of 2045 operation with which you are familiar.

Look at the following POLY expression:

```
RPT 2 (RPT 5 (FOR 10 m SIN(1K*T) FOR 1m 0)
```

In this example, which parenthetically is not a downloaded data example, the RPT 2 is handled by loop counter 0, 0 and the RPT 5 is serviced by loop counter 0, 1.

4. Constant Value: This is used only when Segment Type = 0, 2. Otherwise 0, 0, 0, 0 must be sent.

#### DATA:

This is the actual waveform data point information which is sent as ASCII code corresponding to the 16 bit integer value for the particular data point. This data is sent in a HIGH BYTE, LOW BYTE sequence. Note that the LOW BYTE will always contain zeros, which can be ignored if you have no need for a 16 bit word.

DATA is interpreted for the Analog Output as offset binary. Therefore, the HEX value of the downloaded data corresponds to the Analog output level as follows:

FFFF maximum positive value

—

—

8100 one LSB above mid value

8000 mid value (zero volts)

7FFF one LSB below mid value

—

—

0000 maximum negative value

#### CLOSING DATA HEADER:

This is required to end a download sequence and consists of the same format as used in the data header with segment type = 0, 3. See the Data Header subsection for more detail.

## IEEE FLOATING POINT NUMERICAL CONVERSIONS:

---

As previously discussed, certain data in the Main Header is to be in standard IEEE Floating Point Format. If your host computer does not have a conversion algorithm, and in particular if you are using BASIC we provide the short BASIC program entitled IE3FP.BAS (included in this Section) which you can incorporate into your control program to provide these conversions as required.

### Common Floating Point Values

Sent left to right, one byte per hex pair in string

**Table G-1: Scaling and Offset Values**

REAL	HEX	REAL	HEX
5.0	40A00000	2.5	40200000
4.9	409CCCCD	2.4	4019999A
4.8	4099999A	2.3	40133333
4.7	40966666	2.2	400CCCCD
4.6	40933333	2.1	40066666
4.5	40900000	2.0	40000000
4.4	408CCCCD	1.9	3FF33333
4.3	4089999A	1.8	3FE66666
4.2	40866666	1.7	3FD9999A
4.1	40833333	1.6	3FCCCCCD
4.0	40800000	1.5	3FC00000
3.9	4079999A	1.4	3FB33333
3.8	40733333	1.3	3FA66666
3.7	406CCCCD	1.2	3F99999A
3.6	40666666	1.1	3F8CCCCD
3.5	40600000	1.0	3F800000
3.4	4059999A	0.9	3F666666
3.3	40533333	0.8	3F4CCCCD
3.2	404CCCCD	0.7	3F333333
3.1	40466666	0.6	3F19999A
3.0	40400000	0.5	3F000000
2.9	4039999A	0.4	3ECCCCCD
2.8	40333333	0.3	3E99999A
2.7	402CCCCD	0.2	3E4CCCCD
2.6	40266666	0.1	3DCCCCCD

**Table G-2: Clock Values**

REAL	HEX	REAL	HEX
1.25nS	30ABCC77	5uS	36A7C5AC
2.5nS	312BCC77	10uS	3727C5AC
5nS	31ABCC77	100uS	38D1B717
10nS	322BCC77	1mS	3A83126F
20nS	32ABCC77	2mS	3B03126F
50nS	3356BF95	5mS	3BA3D70A
100nS	33D6BF95	10mS	3C23D70A
200nS	3456BF95	20mS	3CA3D70A
500nS	350637BD	50mS	3D4CCCCD
1uS	358637BD	100mS	3DCCCCCD
2uS	360637BD	1S	3F800000

**Table G-3: Frequency and Filter Values**

400MHz	4DBEBC20	2MHz	49F42400
200MHz	4D3EBC20	1MHz	49742400
100MHz	4CBEBC20	500KHz	48F42400
50MHz	4C3EBC20	200KHz	48435000
20MHz	4B989680	100KHz	47C35000
10MHz	4B189680	50KHz	47435000
5MHz	4A989680	20KHz	469C4000

**SAMPLE DATA DOWNLOAD PROGRAMS:**

There are four (4) sample data download programs included in this section. Examination of these programs will provide a better understanding of the material presented in this section. In fact you might want to actually enter these programs on your host controller and try downloading one or more of them to your 2045 to gain a better feeling for the remote programming dynamics.

These sample programs are written in BASIC and were developed for use on an IBM PC XT (or compatible) with a CEC (Capital Equipment Corporation, Burlington, Ma.) IEEE-488 interface board model number 01000-00200. This interface board utilizes a TI (Texas Instruments) 9914 bus interface device. You may experience certain bus handshake difficulties with other interface hardware.

RPTTEST0.BAS, RPTTEST1.BAS and RPTTEST2.BAS illustrate the use of REPEAT features using both end-shot and nested loops with loop counters 0 and 1. CONSTANT.BAS shows the use of memory compression when a d.c. (constant) value is to be loaded into a minimum amount of output memory space.

**RPTTEST0.BAS**

```

10 CLS
20 'Program to send a DATA set to the Data 2000/2020 or 2040/2045.
30 '
40 INIT%=0 : XMIT%=3 : RECV%=6 : SEND%=9 : SPOLL%=12 : PPOLL%=15 : ENTER%=21
50 D20XX%=16 : PCADDR%=21 : SYSCONT%=0
60 '
70 DEF SEG = &HC000
80 CALL INIT (PCADDR%,SYSCONT%)
90 '
100 ' Place Data 20xx in remote DATA mode.
110 '
120 A$ = SPACE$(255) : A$ = "DATA"
130 CALL SEND (D20XX%,A$,STATUS%)
140 '
150 ' Send the main data header
160 '
170 A$ = "56 81 183 23 0 0 0 0 64 0 0 0 0 0 0 0 "
180 A$ = A$ + "73 244 36 0 76 62 188 32 0 0 0 0 0 0 0 20"
190 GOSUB 600
200 '
210 ' Start RPT counter #0 with a count of 5.
220 ' RPT 5(...)
230 '
240 A$ = "0 5 0 0 0 5 0 0 0 0"
250 GOSUB 600
260 '
270 ' Inform Data 20xx that 20 data points are to follow:
280 '
290 A$ = "0 1 0 0 0 20 0 0 0 0"
300 GOSUB 600
310 '
320 ' Send the 20 data points:
330 '
340 A$ = "0 16 13 135 26 255 40 118 53 238 67 101 80 221 94 85 107 204 "
350 A$ = A$ + "121 68 134 187 148 51 161 170 175 34 188 154 202 17 215 "
360 A$ = A$ + "137 229 0 242 120 255 240"
370 GOSUB 600
380 '
390 ' Inform Data 20xx that 20 more data points are to follow:

```

*continued on next page...*

```
400 '  
410 A$ = "0 1 0 0 0 20 0 0 0 0"  
420 GOSUB 600  
430 '  
440 ' Send the 20 data points:  
450 '  
460 A$ = "128 0 167 136 203 51 231 128 249 173 255 240 249 173 231 128 "  
470 A$ = A$ + "203 51 167 136 128 0 88 120 52 205 24 128 6 83 0 16 6 "  
480 A$ = A$ + "83 24 128 52 205 88 120"  
490 GOSUB 600  
500 '  
510 ' Signal the end of the waveform (and the close of RPT loop #0)  
520 '  
530 A$ = "0 3 0 0 0 20 0 0 0 0"  
540 GOSUB 600  
550 END  
560 '  
570 '  
580 ' Transmit routine  
590 '  
600 CALL XMIT(A$,STATUS%)  
610 RETURN
```

## RPTTEST1.BAS

```

10 CLS
20 'Program to send a DATA set to the Data 2000/2020 or 2040/2045.
30 '
40 INIT%=0 : XMIT%=3 : RECV%=6 : SEND%=9 : SPOLL%=12 : PPOLL%=15 : ENTER%=21
50 D20XX%=16 : PCADDR%=21 : SYSCONT%=0
60 '
70 DEF SEG = &HC000
80 CALL INIT (PCADDR%,SYSCONT%)
90 '
100 ' Place Data 20xx in remote DATA mode.
110 '
120 A$ = SPACE$(255) : A$ = "DATA"
130 CALL SEND (D20XX%,A$,STATUS%)
140 '
150 ' Send the main data header
160 '
170 A$ = "56 81 183 23 0 0 0 0 64 0 0 0 0 0 0 0 "
180 A$ = A$ + "73 244 36 0 76 62 188 32 0 0 0 0 0 0 0 20"
190 GOSUB 660
200 '
210 ' Start RPT counter #1 with a count of 5.
220 ' RPT 5(...)
230 '
240 A$ = "0 5 0 1 0 5 0 0 0 0"
250 GOSUB 660
260 '
270 ' Inform Data 20xx that 20 data points are to follow:
280 '
290 A$ = "0 1 0 0 0 20 0 0 0 0"
300 GOSUB 660
310 '
320 ' Send the 20 data points:
330 '
340 A$ = "0 16 13 135 26 255 40 118 53 238 67 101 80 221 94 85 107 204 "
350 A$ = A$ + "121 68 134 187 148 51 161 170 175 34 188 154 202 17 215 "
360 A$ = A$ + "137 229 0 242 120 255 240"
370 GOSUB 660
380 '
390 ' End RPT counter #1.

```

*continued on next page...*

```
400 ' RPT 5(...)
410 '
420 A$ = "0 6 0 1 0 5 0 0 0 0"
430 GOSUB 660
440 '
450 ' Inform Data 20xx that 20 more data points are to follow:
460 '
470 A$ = "0 1 0 0 0 20 0 0 0 0"
480 GOSUB 660
490 '
500 ' Send the 20 data points:
510 '
520 A$ = "128 0 167 136 203 51 231 128 249 173 255 240 249 173 231 128 "
530 A$ = A$ + "203 51 167 136 128 0 88 120 52 205 24 128 6 83 0 16 6 "
540 A$ = A$ + "83 24 128 52 205 88 120"
550 GOSUB 660
560 '
570 ' Signal the end of the waveform (and the close of RPT loop #0)
580 '
590 A$ = "0 3 0 0 0 20 0 0 0 0"
600 GOSUB 660
610 END
620 '
630 '
640 ' Transmit routine
650 '
660 CALL XMIT(A$,STATUS%)
670 RETURN
```

## RPTTEST2.BAS

```

10 CLS
20 'Program to send a DATA set to the Data 2000/2020 or 2040/2045.
30 '
40 INIT%=0 : XMIT%=3 : RECV%=6 : SEND%=9 : SPOLL%=12 : PPOLL%=15 : ENTER%=21
50 D20XX%=16 : PCADDR%=21 : SYSCONT%=0
60 '
70 DEF SEG = &HC000
80 CALL INIT (PCADDR%,SYSCONT%)
90 '
100 ' Place Data 20xx in remote DATA mode.
110 '
120 A$ = SPACE$(255) : A$ = "DATA"
130 CALL SEND (D20XX%,A$,STATUS%)
140 '
150 ' Send the main data header
160 '
170 A$ = "56 81 183 23 0 0 0 0 64 0 0 0 0 0 0 0"
180 A$ = A$ + "73 244 36 0 76 62 188 32 0 0 0 0 0 0 0 20"
190 GOSUB 720
200 '
210 ' Start RPT counter #0 with a count of 5.
220 ' RPT 5(...)
230 '
240 A$ = "0 5 0 1 0 5 0 0 0 0"
250 GOSUB 720
260 '
270 ' Start RPT counter #1 with a count of 5.
280 ' RPT 5(RPT 5(...) ...)
290 '
300 A$ = "0 5 0 1 0 5 0 0 0 0"
310 GOSUB 720
320 '
330 ' Inform Data 20xx that 20 data points are to follow:
340 '
350 A$ = "0 1 0 0 0 20 0 0 0 0"
360 GOSUB 720
370 '
380 ' Send the 20 data points:
390 '

```

*continued on next page...*



```

400 A$ = "0 16 13 135 26 255 40 118 53 238 67 101 80 221 94 85 107 204 "
410 A$ = A$ + "121 68 134 187 148 51 161 170 175 34 188 154 202 17 215 "
420 A$ = A$ + "137 229 0 242 120 255 240"
430 GOSUB 720
440 '
450 ' End RPT counter #1.
460 ' RPT 5(...)
470 '
480 A$ = "0 6 0 1 0 5 0 0 0 0"
490 GOSUB 720
500 '
510 ' Inform Data 20xx that 20 more data points are to follow:
520 '
530 A$ = "0 1 0 0 0 20 0 0 0 0"
540 GOSUB 720
550 '
560 ' Send the 20 data points:
570 '
580 A$ = "128 0 167 136 203 51 231 128 249 173 255 240 249 173 231 128 "
590 A$ = A$ + "203 51 167 136 128 0 88 120 52 205 24 128 6 83 0 16 6 "
600 A$ = A$ + "83 24 128 52 205 88 120"
610 GOSUB 720
620 '
630 ' Signal the end of the waveform (and the close of RPT loop #0)
640 '
650 A$ = "0 3 0 0 0 20 0 0 0 0"
660 GOSUB 720
670 END
680 '
690 '
700 ' Transmit routine
710 '
720 CALL XMIT(A$,STATUS%)
730 RETURN

```

## CONSTANT.BAS

```

10 CLS
20 'Program to send a DATA set to the Data 2000/2020 or 2040/2045.
30 '
40 INIT%=0 : XMIT%=3 : RECV%=6 : SEND%=9 : SPOLL%=12 : PPOLL%=15 : ENTER%=21
50 D20XX%=16 : PCADDR%=21 : SYSCONT%=0
60 '
70 DEF SEG = &HC000
80 CALL INIT (PCADDR%,SYSCONT%)
90 '
100 ' Place Data 20xx in remote DATA mode.
110 '
120 A$ = SPACE$(255) : A$ = "DATA"
130 CALL SEND (D20XX%,A$,STATUS%)
140 '
150 ' Send the main data header
160 '
170 A$ = "56 81 183 23 0 0 0 0 64 0 0 0 0 0 0 0 "
180 A$ = A$ + "73 244 36 0 76 62 188 32 0 0 0 0 0 0 0 20"
190 GOSUB 540
200 '
210 ' Inform Data 20xx that 20 data points are to follow:
220 '
230 A$ = "0 1 0 0 0 20 0 0 0 0 "
240 GOSUB 540
250 '
260 ' Send the 20 data points:
270 '
280 A$ = "0 16 13 135 26 255 40 118 53 238 67 101 80 221 94 85 107 204 "
290 A$ = A$ + "121 68 134 187 148 51 161 170 175 34 188 154 202 17 215 "
300 A$ = A$ + "137 229 0 242 120 255 240"
310 GOSUB 540
320 '
330 ' Inform Data 20xx that 20 more data points are to follow:
340 '
350 A$ = "0 1 0 0 0 20 0 0 0 0 "
360 GOSUB 540
370 '
380 ' Send the 20 data points:
390 '

```

*continued on next page...*

```
400 A$ = "128 0 167 136 203 51 231 128 249 173 255 240 249 173 231 128 "  
410 A$ = A$ + "203 51 167 136 128 0 88 120 52 205 24 128 6 83 0 16 6 "  
420 A$ = A$ + "83 24 128 52 205 88 120"  
430 GOSUB 540  
440 '  
450 ' Send a constant value of 40 points.  
460 '  
470 A$ = "0 2 0 0 0 40 200 0 0 0 "  
480 GOSUB 540  
490 END  
500 '  
510 '  
520 ' Transmit routine  
530 '  
540 CALL XMIT(A$,STATUS%)  
550 RETURN
```

## IEF3FP.BAS

```

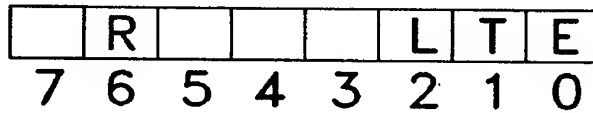
10 'IE3FP.BAS
20 '
30 'TO TEST IEEE FLOATING POINT CONVERSION ROUTINE
40 '
50 '
60 '
70 INPUT "ENTER THE NUMBER".FP#
80 GOSUB 240
90 GOSUB 150
100 PRINT " "
110 PRINT " "
120 GOTO 70
130 END
140 '
150 A=B1%:B=B2%:C=B2%:D=B4%
160 IF A=0 AND B=0 AND C=0 AND D=0 THEN FP=0:GOTO 210
170 S=(A AND 128)/128
180 E=(A AND 127)*2+(B AND 128)/128-127
190 M#=((((B AND 127)+128)*2^16+C*2^8+D)/2^23
200 FP#=(-1)^S*2^E*M#
210 PRINT "THE NUMBER IS ";FP#
220 RETURN
230 '*****CONVERSION ROUTINE*****
240 SIGNFP%=SGN(FP#)
250 IF SIGNFP%=-1 THEN FP#=FP#*(-1) ELSE SIGNFP%=0
260 IF FP#=0 THEN B1%=0 AND B2%=0 AND B3%=0 AND B4%=0:HEAD$="0 0 0 0 ":RETURN
270 EXPON#=LOG(FP#)*1.442695
280 IEXPON%=INT(EXPON#)
290 MANT#=FP#*(2^(23-IEXPON%))-2^23+1
300 IF MANT#8388607! THEN IEXPON%=IEXPON%+1:GOTO 290
310 B=INT(MANT#/65536!)
320 C1#=MANT#-B*65536!
330 C=INT(C1#/256)
340 D=INT(C1#-C*256)-1
350 B4%=D
360 B3%=C
370 IEXPON%=IEXPON%+127
380 B2%=(IEXPON% AND 1)*128+B
390 IF SIGNFP%=-1 THEN SIGNFP%=1
400 B1%=(IEXPON% AND 254)/2+SIGNFP%*128
410 HEAD$=STR$(B1%)+""+STR$(B2%)+""+STR$(B3%)+""+STR$(B4%)+""
420 PRINT " BYTES ARE ":HEAD$
430 RETURN

```

# SRQs and Status Bits

## 1. SRQ Generation and The Status Byte

The 2045 is capable of generating a service request (SRQ) on three conditions: (1) when an error occurs (ERRSRQ), (2) when it is ready to receive a command (LISNSRQ), and (3) when the output buffer is loaded (TALKSRQ or OUTSRQ). A corresponding bit in the status byte is set to indicate which condition has occurred. It is possible to have an SRQ generated by multiple conditions, in which case multiple bits will be set in the status byte. If a second SRQ is generated before the first can be serviced, the new status byte is the logical OR of the original and updated status. Figure 1 shows the status byte definition.



*Figure 1 - Status Byte Definition*

**Bit 6 = Request Service**

1 = indicates 2045 requesting service

0 = indicates 2045 not requesting service

**Bit 3 = Memory Test**

1 = Memory Test in progress

0 = Memory Test not active

*Note: Bit 3 is a status bit only and will not generate an SRQ.*

**Bit 2 = Listen SRQ (LISNSRQ)**

1 = Ready to receive a command

0 = Not ready

**Bit 1 = Talk SRQ (TALKSRQ)**

1 = Talk buffer not empty

0 = Talk buffer empty

**Bit 0 = Error SRQ (ERRSRQ)**

1 = Error has occurred

0 = No error

## 2. Enabling the SRQs

To write an interrupt driven program you must enable the 2045 SRQs and the interrupts on your controller or PC. The Error SRQ (ERRSRQ) is always enabled; the other two (LISNSRQ and TALKSRQ) may be enabled via the OP 488 menu or remote bus command. See Section C for a complete discussion of the OP 488 menu. The mnemonics for enabling the SRQs by the remote command are given below:

TALKSRQ = 1 enables Talk SRQ  
TALKSRQ = 0 disables Talk SRQ (default state)

LISNSRQ = 1 enables Listen SRQ  
LISNSRQ = 0 disables Listen SRQ

The default state for the Listen SRQ may be set via switch position 1 on the rear panel switch assembly. To set the Listen SRQ default state to "ON" set switch 1 to the "UP" position; to set the default state to "OFF" set switch 1 to the "DOWN" position.

Consult the User's Manual for your controller or PC to determine the procedure for enabling its interrupts.

## 3. Servicing an SRQ

Once an SRQ has been generated it may be serviced by a procedure called polling. Polling usually takes place in a subroutine designed to take a given course of action depending on the device which is requesting service and the condition which caused the service request. If there is only one device on the bus the type of poll which takes place is called a serial poll; if there are multiple devices on the bus the type of poll which is usually performed is called a parallel poll.

*Note:* The 2045 does not support parallel polling; however, this does not imply that the 2045 must be the only device on the bus when using the SRQs. It is possible to perform a serial poll of each device when there are multiple devices present.

A typical service routine for the 2045 would do the following:

- 1) Conduct a serial poll. This clears the service request line on the GPIB interface and returns the 2045's status byte.
- 2) Test bit 6 of the status byte to insure that the polled device actually requested service.
- 3) Test the assigned bits in the status byte to determine the nature of the service request.
- 4) Re-enable the interrupts on the controller or PC.
- 5) Branch to another subroutine or back to the main program and provide the required service.

The exact syntax of the serial poll command is controller dependent and will vary depending on which GPIB interface card you have in your controller or PC. A typical serial poll command in HP BASIC will have the following form:

Stat = SPOLL(addr)

where: Stat = the value of the 2045 status byte  
addr = the GPIB address of the 2045

#### 4. Programming Hints

When programming the 2045 give careful thought to the task you want to accomplish before enabling the SRQs. Generally speaking, using all three interrupts simultaneously is not recommended. Each SRQ was designed with specific objectives in mind and will best serve the user when used in the correct manner. Experience has shown that code written without consideration of these objectives may often prove to be frustrating and difficult to debug. Listed below are some recommended guidelines and suggestions for using the 2045 SRQs:

1. When logging errors use only ERRSRQ and LISNSRQ. The error mode is always set to "on request" for the GPIB, so to read an error send the ERROR command after LISNSRQ has been received. Since the LISNSRQ may or may not be sent concurrently with the ERRSRQ it is necessary to explicitly test for it before issuing the ERROR command.
2. LISNSRQ, with the default condition set to "ON", is especially useful in ATE systems to tell the controller when the 2045's power-up initialization is complete and to indicate that the 2045 is finished calculating an expression.
3. TALKSRQ should normally be used only when issuing the DIR or DUMP commands.

## Key Code Reference

In addition to the mnemonics, remote control can simulate direct key presses by sending  $\text{KEY} = n$  where  $n$  is a number from 0 to 63. Each key on the front panel of a 20XX is assigned to one of these numbers, as shown in figure G-1 and as indicated in the table below. See also the second table listed alphabetically by key name.

The key codes are suggested as a means of simulating certain front panel operations, such as scrolling through menus with the 20XX display active by using FIELD key code,  $\text{KEY} = 1$ , several times in succession.

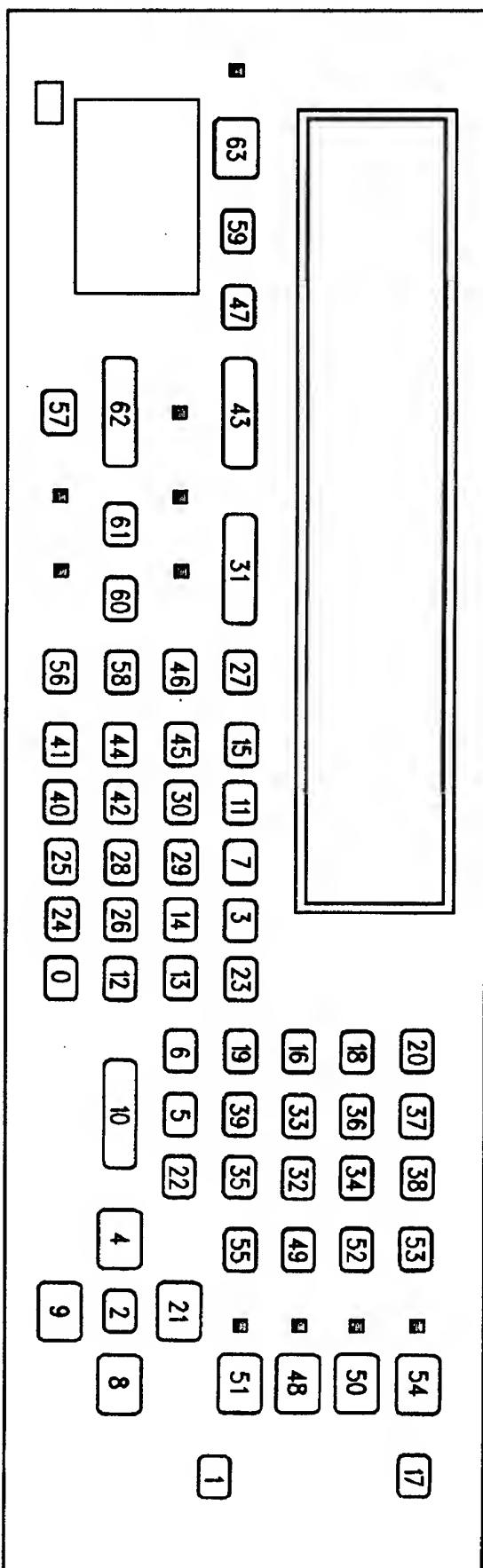
Note: In the first table, where two key names appear together (such as SIN and ARCSIN), this means that both functions are served by the same key or key code, the second function invoked by preceding it with code 6 for the "2nd" key.

Key number	Key name	Key number	Key name
0	FILT	32	9
1	FIELD	33	8
2	FAST	34	6 or m
3	(	35	EE or e
4	$\leftarrow$	36	5 or u
5	F	37	2 or M
6	2nd	38	3 or =
7	TO or SGN	39	0 or II
8	$\Rightarrow$	40	OFST
9	$\downarrow$	41	S + N
10	SPACE	42	$\gamma^x$
11	AT or ABS	43	ENTER
12	T	44	TAN or ARCTAN
13	t	45	COS or ARCCOS
14	LN	46	STO
15	SIN or ARCSIN	47	NO
16	7 or OP	48	TRIANGLE WAVE
17	FORMAT	49	—
18	4 or n	50	SINE WAVE
19	.	51	NOISE
20	1 or K	52	x
21	$\uparrow$	53	$\div$
22	CE or CLR	54	SQUARE WAVE
23	)	55	+
24	CLK	56	DEL
25	MARK	57	MANUAL TRIG
26	LOG	58	RCL
27	DIR	59	YES
28	INT	60	REM/LOC
29	FOR	61	CYC/RAD
30	RPT	62	RESTORE
31	HELP	63	R/S



For quick cross reference and ease of use, a second table below lists key names alphabetically followed by the relevant key number code. All of the "2nd" shifted functions are shown as 6, X to indicate that a 6 must be sent to activate the "2nd" key before the function can be activated.

Key name	Key number	Key name	Key number
0	39	[LEFT ARROW] ⇐	4
1	20	[LEFT PARENTHESIS] (	3
2	37	LN	14
3	38	LOG	26
4	18	MANUAL TRIG	57
5	36	MARK	25
6	34	[MEGA] M	6, 37
7	16	[MICRO] u	6, 36
8	33	[MILLI] m	6, 34
9	32	[MULTIPLY] x	52
ABS	6, 11	[NANO] n	6, 18
[ADD] +	55	NO	47
ARCCOS	6, 45	NOISE	51
ARCSIN	6, 15	OFST	40
ARCTAN	6, 44	OP	6, 16
AT	11	[PI] $\Pi$	6, 39
CE	22	R/S	63
CLK	24	RCL	58
CLR	6, 22	REM/LOC	60
COS	45	RESTORE	62
CYC/RAD	61	[RIGHT ARROW] ⇒	8
[DECIMAL POINT] .	19	[RIGHT PARENTHESIS] )	23
DEL	56	RPT	30
DIR	27	[SECOND] 2nd	6
[DIVIDE] ÷	53	SGN	6, 7
[DOWN ARROW] ↓	9	SIN	15
e	6, 35	SINE WAVE	50
EE	35	S + N	41
ENTER	43	SPACE	10
EQUAL	6, 38	SQUARE WAVE	54
[EXPONENT] Y <sup>x</sup>	42	STO	46
F	5	[SUBTRACT] −	49
FAST	2	TAN	44
FIELD	1	[TIME, GLOBAL] T	12
FILT	0	[TIME, LOCAL] t	13
FOR	29	TO	7
FORMAT	17	TRIANGLE WAVE	48
HELP	31	[UP ARROW] ↑	21
INT	28	YES	59
[KILO] K	6, 20		



### Figure G-1



# MNEMONIC FUNCTIONAL LISTING

This is a list of mnemonics for commands, modifiers and math operators shown alphabetically within categories with brief descriptions. The mnemonics fall into three general categories:

## 1. COMMANDS

Commands perform an immediate execution function.

## 2. MODIFIERS

Modifiers are parameter labels that are set equal to a value. They do not execute a function as do commands, but are used to change operating parameters which affect how some commands are executed. There are two types of modifiers:

A. Mode Independent: These modifiers operate with any currently active operating mode.

B. Mode Dependent: These are modifiers that operate differently or not at all depending on which of the three operating modes is currently active. If a modifier is not valid for a certain mode, it is not shown under the mode sub-heading in the following list. The modes are:

1. Polynomial mode (POLY)
2. Standard Function mode (SFM)
3. Modified mode (MOD)

## 3. MATH OPERATORS

Math operators are mathematical functions, constants, and directive statements.

## COMMANDS

CLR	CLEARs EDIT BUFFER
CONFIG	REMOTEly DISPLAYs INSTALLED OPTIONS
CYC	ENABLEs CYCLES PER SECOND MODE
DATA	ENABLEs REMOTE DATA DOWNLOAD MODE
DEL	DELETES SPECIFIED WAVEFORM FROM NON-VOLATILE MEMORY
DIR	DISPLAYs NAMES OF STORED FILES
DUMP	DISPLAYs NAMES OF STORED FILES AND CONTENTS
ENTER	ENABLEs COMPUTATION OR EXECUTES PARAMETER CHANGE
ERROR	DISPLAYs NEXT ERROR IN QUEUE
FIELD	DUPLICATES PRESSING OF FIELD KEY
INIT	DELETES ALL FILES IN NON-VOLATILE RAM
KEY	PRESSES FRONT PANEL KEYS BY NUMBER CODE
LOC	ENABLEs LOCAL CONTROL
MEMSTS	RETURNS THE STATUS OF THE MEMORY READ/WRITE TEST
MOD	ENTERS MODIFIED MODE
OP0	ENTERS SCOPE DRAW MODE
OP4	SETS OPERATION SETUP PARAMETERS
OP7	DISPLAYs LIST OF SPECIAL FEATURES
OP232	SETS SERIAL COMMUNICATIONS PARAMETERS
OP488	SETS PARALLEL COMMUNICATIONS PARAMETERS
OP1510	ENTERS TABLET DRAW MODE
OP 6000	RETRIEVES CAPTURED WAVEFORM FROM D6000 SERIES WAVEFORM ANALYZER
OP 6001	CHANGES END POINT OF WAVEFORM ALREADY RETRIEVED WITH OP 6000.
OP 9999	DELETES ALL SAVED FILES FROM NON-VOLATILE MEMORY
POLY	ENABLEs POLYNOMIAL MODE
PURGE	DELETES ALL WAVEFORMS FROM OUTPUT MEMORY
RAD	ENABLEs RADIANs PER SECOND MODE
RCL	RECALLs SPECIFIED WAVEFORM
REM	ENABLEs REMOTE CONTROL, LOCKS OUT FRONT PANEL
RES	EXITS OP MODE OR MODIFIED MODE, SUSPENDs COMPUTATION
RESET	SIMULATES POWER OFF THEN ON
RSET	SAME AS RESET
RESTORE	SAME AS RES
RUN	ENABLEs OUTPUT
SNOI	SELECTs NOISE STANDARD FUNCTION MODE
SSIN	SELECTs SINE WAVE STANDARD FUNCTION MODE
SSQR	SELECTs SQUARE WAVE AND PULSE STANDARD FUNCTION MODE
STOP	DISABLEs OUTPUT
STRI	SELECTs TRIANGLE AND SAWTOOTH WAVE STANDARD FUNCTION MODE

**MODE INDEPENDENT MODIFIERS**

BADR	SELECTS BUS ADDRESS
BLOCK	DEFINES MAXIMUM SIZE OF UPLOADED DATA IN BYTES
BUS6000	SELECTS THE DATA 6000 BUS ADDRESS
CLKLEV	SETS CLOCK LEVEL
CLKSRC	SELECTS CLOCK SOURCE
ECHO	ENABLES/DISABLES DISPLAY OF REMOTE COMMANDS
EOIOUT	ENABLES/DISABLES MESSAGE TERMINATOR
ERRM	SELECTS MODE OF ERROR REPORTING
ERRSRQ	ENABLES/DISABLES THE ERROR SERVICE REQUEST
FASTIO	ENABLES/DISABLES DIRECT MEMORY ACCESS
GET	ENABLES/DISABLES GROUP EXECUTE TRIGGER
LISNSRQ	ENABLES/DISABLES READY TO RECEIVE SERVICE REQUEST
OUTSEL	SELECTS THE ACTIVE SIGNAL OUTPUT, A OR B
PROMPT	ENABLES/DISABLES PROMPT ON RS-232 REMOTE CONTROLLER SCREEN
TALKSRQ	ENABLES/DISABLES READY TO SEND SERVICE REQUEST
TGTPNTS	SETS THE APPROXIMATE NUMBER OF POINTS TO BE COMPUTED
TRIGLEV	SETS TRIGGER LEVEL

**MODE DEPENDENT MODIFIERS*****POLY:***

CLK	SETS CLOCK RATE, OVERRIDES AUTO CLOCK RATE
FILT	SELECTS FREQUENCY OF WAVEFORM FILTER
MARK	SETS OCCURRENCE OF MARKER PULSE IN SECONDS
NAMP	SETS AMPLITUDE OF NOISE IN VOLTS RMS
NBW	SELECTS BANDWIDTH OF NOISE
OFST	SETS VERTICAL OFFSET

***SFM:***

AMP	SETS AMPLITUDE OF WAVEFORM
DLY	SETS DELAY BETWEEN SYNC OUT AND SQUARE OR TRIANGLE WAVE
DUTY	SELECTS PERCENTAGE OF SQUARE WAVE PERIOD AT MAX AMP
FILT	SELECTS FREQUENCY OF WAVEFORM FILTER
FREQ	SETS FREQUENCY OF WAVEFORM
HIGH	SETS MAX VALUE OF SQUARE WAVE
LOW	SETS MIN VALUE OF SQUARE WAVE
MARK	SETS OCCURRENCE OF MARKER PULSE
MPTS	SETS OCCURRENCE OF MARKER PULSE AS DEFINED BY NUMBER OF POINTS
NAMP	SETS AMPLITUDE OF NOISE IN VOLTS RMS
NBW	SELECTS BANDWIDTH OF NOISE
NCLK	SETS CLOCK RATE FOR NOISE MODE
NOISE	ENABLES/DISABLES NOISE
NSEED	SELECTS STARTING POINT OF NOISE FROM RANDOM VALUE STRING
OFST	SETS VERTICAL OFFSET
PER	SETS PERIOD OF WAVEFORM
PHS	SETS PHASE ANGLE BETWEEN SYNC OUT AND SINE WAVE

PLSW        SETS PULSE WIDTH  
 SYM        SETS SYMMETRY OF SINE OR TRIANGLE WAVE  
 TRIGMODE   SELECTS TRIGGER MODE

**MOD:**

FILT        SELECTS FREQUENCY OF WAVEFORM FILTER  
 MARK       SETS OCCURRENCE OF MARKER PULSE  
 MPTS       SETS OCCURRENCE OF MARKER PULSE AS DEFINED BY NUMBER OF POINTS  
 CLK        SETS CLOCK RATE, OVERRIDES AUTO CLOCK RATE  
 NOISE      ENABLES/DISABLES NOISE  
 OFST       SETS VERTICAL OFFSET  
 SN        SETS SIGNAL PLUS NOISE AMPLITUDE LEVEL  
 TRIGM      SELECTS TRIGGER MODE

**MATH OPERATORS**

ABS        DEFINES FUNCTION WHICH EVALUATES ABSOLUTE VALUE OF EXPRESSION  
 ARCCOS    DEFINES FUNCTION WHICH EVALUATES ARCCOSINE OF EXPRESSION  
 ARCSIN    DEFINES FUNCTION WHICH EVALUATES ARCSINE OF EXPRESSION  
 ARCTAN    DEFINES FUNCTION WHICH EVALUATES ARCTANGENT OF EXPRESSION  
 AT        INTERPOLATES DEFINED OUTPUT AT GLOBAL TIME T  
 COS       COMPUTES COSINE OF EXPRESSION  
 e        CONSTANT = 2.7182818  
 -e        CONSTANT = -2.7182818  
 EXP       DEFINES FUNCTION WHICH RAISES Y TO X POWER  
 FOR       GENERATES DEFINED OUTPUT WITH DURATION = LOCAL TIME t  
 INT       DEFINES FUNCTION WHICH EVALUATES INTEGRAL OF EXPRESSION  
 LN        DEFINES FUNCTION WHICH EVALUATES NATURAL LOG OF EXPRESSION  
 LOG       DEFINES FUNCTION WHICH EVALUATES COMMON LOG OF EXPRESSION  
 PI        CONSTANT = 3.1415927  
 pi        LOWER CASE OPTION OF PI  
 -PI       CONSTANT = -3.1415927  
 -pi       LOWER CASE NEGATIVE OF PI  
 RPT       REPEATS SEGMENTS, PROVIDES N-SHOT  
 SGN       DEFINES FUNCTION WHICH EVALUATES SIGNUM OF EXPRESSION  
 SIN       DEFINES FUNCTION WHICH EVALUATES SINE OF EXPRESSION  
 T        GLOBAL (TOTAL) TIME VARIABLE  
 t        LOCAL TIME VARIABLE  
 TAN       DEFINES FUNCTION WHICH EVALUATES TANGENT OF EXPRESSION  
 TO        STOPS DEFINED OUTPUT AT GLOBAL TIME T  
 +        ADDITION SYMBOL  
 -        SUBTRACTION SYMBOL  
 \*        MULTIPLICATION SYMBOL  
 /        DIVISION SYMBOL

## MNEMONIC GENERAL REFERENCE

This subsection contains all mnemonic commands, modifiers and math operators used for control from a remote host. The mnemonics are listed in alphabetical order with functional definitions and syntax examples.

Mnemonics accompanied by "key" in parentheses means there is a front panel key serving the same function, but not necessarily identified by the same letters or symbols.

**ABS (key)**      Type: Math Operator

**Function:** ABS is the abbreviation for Absolute Value, where all values are considered only in terms of positive magnitude. All negative values become positive, all positive values remain unchanged.

**Syntax:** FOR 1m ABS(SIN(1K\*t))

**Note:** In the above syntax example, the resulting waveform will look the same as an AC sine wave that has been run through a full wave rectifier.

**AMP**      Type: MD Modifier

**Function:** Allows setting of waveform peak amplitude in volts.

**Syntax:** AMP = 1.00027 ('A' default), = 0.5 ('B' default)

**Note:** AMP can be zero, but negative values will not be accepted. AMP works only with Standard Function mode. AMP combined with OFST has an alternate format of HIGH and LOW.

**ARCCOS (key)**      Type: Math Operator

**Function:** Defines function which evaluates ArcCosine of expression.

**Syntax:** ARCCOS(1K\*t)



**ARCSIN (key)**      Type: Math Operator

Function: Defines function which evaluates ArcSine of expression.

Syntax: ARCSIN(1K\*t)

**ARCTAN (key)**      Type: Math Operator

Function: Defines function which evaluates ArcTangent of expression.

Syntax: ARCTAN(1K\*t)

**AT (key)**      Type: Math Operator

Function: Produces a linearly interpolated waveform segment where the starting voltage is the last point calculated before the AT segment begins. Duration of the segment and the final voltage are defined in the AT statement.

Syntax: AT 1m 3 (final voltage is 3)

Note: AT statements are referenced to global time (big T), from the beginning of the polynomial, which may or may not be the beginning of the AT statement. AT statements can be used only with constants.

**BADR**      Type: MI Modifier

Function: Sets the IEEE-488 (GPIB) bus address.

Syntax: BADR = n, where n = 0 to 31

Note: The default Bus Address setting is determined by the rear panel DIP switches.

**CLK (key)**      Type: MD Modifier

Function: Used to override calculated output clock rate.

Syntax: CLK = 40n

Note: CLK works only with POLY and MODIFIED modes.

**CLKLEV** Type: MI Modifier

Function: Selects one of two clock threshold levels of the output clock when using **EXTERNAL CLOCK**.

Syntax: CLKLEV = 0 (for 0V), = 1 (for 1.5V, default)

**CLKSRC** Type: MI Modifier

Function: Selects internal or external clock source.

Syntax: CLKSRC = 0 (for internal, default), = 1 (for +external), = 2 (for -external), = 3 (for ECL clock)

**CLR (key)** Type: Command

Function: Clears the edit buffer in POLY mode, also clears numeric only fields individually in standard function mode.

Syntax: CLR

Note: 2nd shifted CE key.

**CONFIG** Type: Command

Function: Sent from a remote terminal, will display list of special features on terminal screen.

Syntax: CONFIG

**COS (key)** Type: Math Operator

Function: Defines a function in a polynomial which evaluates the cosine of the expression.

Syntax: COS(1K\*t)

Note: If in CYC mode, the the argument is evaluated in cycles. If in RAD mode, the argument is evaluated in radians. If a phase term is used, such as COS(1K\*t + .25) in CYC mode, it has an equivalent in RAD mode which in this example would be COS(2\*PI\*1K\*t + PI/2).

**CYC (key)** Type: Command

**Function:** Determines trigonometric arguments will be interpreted as cycles per second.

**Syntax:** CYC (default setting of CYC/RAD key)

**Note:** See note under COS.

**DATA** Type: Command

**Function:** Invokes the remote data download mode, required before data can be accepted.

**Syntax:** DATA

**DEL (key)** Type: Command

**Function:** Deletes specified waveform from memory. Unlike key entry at front panel, this mnemonic does not do a courtesy check before deletion.

**Syntax:** DEL F2

**DIR (key)** Type: Command

**Function:** Returns listing of stored filenames only. To return equations, see DUMP.

**Syntax:** DIR

**DLY** Type: MD Modifier

**Function:** In square wave mode, used to select the amount of delay time (in seconds) between the leading edge of the SYNC OUT pulse and the leading edge of the square wave or pulse at the output connector. In triangle wave mode, measured to the center of the leading slope at output.

**Syntax:** DLY = 35u (default = 0)

**Note:** The DLY parameter for both SSQR and STRI will be changed to the new value when sent from remote. DLY works only with Standard Function mode.

**DUMP**      Type: Command

Function: Returns listing of stored filenames and their corresponding equations.

Syntax: DUMP

**DUTY**      Type: MD Modifier

Function: Used to select the amount of time (in percent of one cycle or period) that a square wave will stay high.

Syntax: DUTY = 50

Note: When DUTY cycle is changed, SYM (symmetry), which is the related parameter for SSIN (standard function Sine) and STRI (standard function Triangle) will also change. This becomes apparent if you switch to these other modes. The alternate format of DUTY in square wave mode is PLSW.

**e (key)**      Type: Math Operator  
also -e

Function: 'e' is used to represent the constant value 2.7182818 in polynomial equations. '-e' is the negative value of this constant.

Syntax: FOR 10m  $(200 * t) * e^{(-1K * t)}$

Note: The value of 'e' is equal to the base of the natural log function, LN. Key is 2nd shifted EE.

**ECHO**      Type: MI Modifier

Function: When enabled, ECHO will cause the display to monitor the action resulting from remote commands. In this way, it is similar to local operation from the front panel keys.

Syntax: ECHO = 0 (to disable, default), = 1 (to enable)

**ENTER**      Type: Command

Function: Enables computation if in STOP mode or executes computation if in RUN mode of polynomial or standard function currently shown on display. Also executes a change of a menu parameter (modifier) value. This command duplicates pressing the ENTER key.

Syntax: ENTER

**EOIOUT** Type: MI Modifier

**Function:** Message terminator, 'End Or Identify' word. Used to indicate the end of a multiple byte transmission.

**Syntax:** EOIOUT = 0 (to disable), = 1 (to enable, default)

**Note:** When ON, a necessary condition when using DMA, EOI is expected on receipt and supplied on transmission. When OFF, EOI is optional on receipt and not supplied on transmission.

**ERRM** Type: MI Modifier

**Function:** Used to set error reporting mode to 'immediate' or 'on request' when used with RS-232 option. If communications are over the GPIB (IEEE-488) bus, errors are reported by request only, using error SRQs, disregarding the setting of ERRM.

**Syntax:** ERRM = 0 (immediate, default), = 1 (on request)

**Note:** ERRM = 'immediate' is useful when using a dumb terminal for error logging.

**ERROR** Type: Command

**Function:** Displays and clears out the next error in the error queue, advancing all other errors in the queue one space. The error queue can hold up to 16 errors, stored in a FIFO (first in, first out) format. When the queue is full, additional errors cannot be stored and are lost. Used in conjunction with ERRM = 1.

**Syntax:** ERROR

**EXP (key)** Type: Math Operator  
[Y<sup>x</sup>]

**Function:** Defines the function in a polynomial which raises Y to the Xth power in a polynomial expression.

**Syntax:** FOR 1m SIN(1K\*t)^(3)

**Note:** A negative value cannot be raised to a non-integer power, e.g. FOR 1m SIN(1K\*t)^(3.2) will not work because SIN(1K\*t) goes negative during the time interval specified, and a fractional exponent applied to a negative base yields an imaginary number.

**FASTIO** Type: MI Modifier

**Function:** Enables or disables DMA (Direct Memory Access). When enabled, it speeds up data transfer to over 300K bytes/sec. Useful only when DMA option is installed.

**Syntax:** FASTIO = 0 (to disable, default), = 1 (to enable)

**FIELD** Type: Command

**Function:** Duplicates pressing the FIELD key, which advances the cursor one field forward in menu modes.

**Syntax:** FIELD

**FILT (key)** Type: MI Modifier

**Function:** Sends the output waveform through a passive low pass filter, selectable at two different frequencies. If you enter a value in between the set frequencies, the value you enter will remain displayed in your polynomial, but the actual value used will be rounded up or down to the nearest available frequency. The filter frequency actually used can be seen by pressing the RESTORE key. The filters available are 2MHz and 20MHz.

**Syntax:** FILT = 2M, FILT = 20M, or FILT = 50M for no filter

**FOR (key)** Type: Math Operator

**Function:** Specifies a waveform segment duration.

**Syntax:** FOR 1m SIN(1K\*t)

**Note:** FOR statements are referenced to local time (small t), which starts at  $t=0$  whenever the FOR statement begins, not necessarily at the beginning of the polynomial. There can be a number of FOR statements, each with its own 't' segment, e.g. FOR t (x) FOR t (x) FOR t (x) etc., each 'FOR t' starting at 0 and going to its assigned value.

**FREQ** Type: MD Modifier

**Function:** Selects the desired frequency of a standard function waveform.

**Syntax:** FREQ = 100K

**Note:** When this command is sent, the frequency of all standard waveforms is changed regardless of which one is active.

**GET**      Type: MI Modifier

Function: Group Execute Trigger. Upon receipt of a GPIB (IEEE-488) GET signal, waveform output will either stop or start depending on trigger mode. Can be enabled or disabled.

Syntax: GET = 0 or NONE (default), = 1 or TRIGGER

**HIGH**      Type: MD Modifier

Function: Used to select the maximum value in volts of the waveform segment when in Standard Function mode.

Syntax: HIGH = 2.5

Note: HIGH and LOW are alternate formats for AMP and OFST for the SSIN, SSQR, and STRI waveforms. If ECHO is enabled, they are automatically displayed on the screen when the command is sent.

**INIT**      Type: Command

Function: Used to clear non-volatile RAM.

Syntax: INIT

Note: This command will not perform a courtesy check.

**INT (key)**      Type: Math Operator

Function: Defines the function in a polynomial which Evaluates the integral of an expression by boxcar integration.

Syntax: FOR 100m INT(1K\*T)

Note: Key is labeled with integration symbol, not INT.

**KEY**      Type: Command

Function: Used to simulate pressing of keys on front panel.

Syntax: KEY = n where n is a key number in the range 0-63

Note: See Fig. G1 at the end of section G and two pages preceding it for assignment of key numbers.

**LISNSRQ** Type: MI Modifier

**Function:** Enables or disables the Listen service request (SRQ) which indicates readiness to receive data.

**Syntax:** LISNSRQ = 0 (off,default), or = 1 (on)

**LN (key)** Type: Math Operator

**Function:** Defines the function in a polynomial which evaluates the natural log of expression, base 2.7182818.

**Syntax:** FOR 1m LN(10K\*T)

**Note:** Base value of natural log can be called up separately with 'e' key or mnemonic.

**LOC (key)** Type: Command

**Function:** Returns control to local state, enables front panel keys.

**Syntax:** LOC (default setting of REM/LOC key)

**LOG (key)** Type: Math Operator

**Function:** Defines the function in a polynomial which evaluates the common log of expression, base 10.

**Syntax:** FOR 1m LOG(10K\*T)

**LOW** Type: MD Modifier

**Function:** Used to select the minimum value in volts of the waveform segment when in standard function mode.

**Syntax:** LOW = -2.5

**Note:** See HIGH



**MARK (key)**     Type: MD Modifier

**Function:** Used to select the time or number of clock pulses when a marker pulse will go high.

**Syntax:** MARK = time (in seconds)

**Note:** MPTS is the alternate format of MARK.

**MEMSTS**     Type: Command

**Function:** Returns the status of the waveform memory Write/Read test.

**Syntax:** MEMSTS

**Note:** Memory test status is returned as one of four numbers from 0 to 3, interpreted as shown:

- 0 - memory test completed, no errors detected
- 1 - memory test completed, errors detected
- 2 - memory test aborted, no errors detected before termination
- 3 - memory test aborted, errors detected before termination

**MOD (key)**     Type: Command

**Function:** Enters MODIFIED mode.

**Syntax:** MOD

**MPTS**     Type: MD Modifier

**Function:** Alternate form of MARK, MPTS defines marker pulse as number of points.

**Syntax:** MPTS = # points

**NAMP**     Type: MD Modifier

**Function:** Sets the amplitude in volts RMS of generated noise.

**Syntax:** NAMP = 0.1

**NBW**      Type: MD Modifier

**Function:** Used to select NOISE 3dB Bandwidth, in hertz. Will select nearest valid value.

**Syntax:** NBW = 20K

**NCLK**      Type: MD Modifier

**Function:** Used to set the clock rate when in NOISE standard function mode.

**Syntax:** NCLK = 1.00787u

**Note:** NCLK is set independently of CLK; they do not track each other.

**NOISE**      Type: MD Modifier

**Function:** Enables or disables the generation of psuedo-random noise in any of the four standard function modes. Controls the NOISE = ON or HOLD field which appears in all these modes.

**Syntax:** NOISE = 0 (ON), = 1 (HOLD)

**Note:** Noise can be used alone, as when NOISE standard function mode is selected, or added to one of the standard function waveforms being produced. Does not duplicate pressing the NOISE key. For that function, see the mnemonic heading SNOI.

**NSEED**      Type: MI Modifier

**Function:** Determines the starting point of NOISE generation from within a string of random variables.

**Syntax:** NSEED = +/-9,999,999; except NSEED cannot = 0

**Note:** NOISE points are generated in a truly random pattern, but the pattern is repeatable, making the NOISE actually psuedo-random. The integer value that you assign to NSEED designates only a unique and consistent starting point for that repeatable pattern. The integer NSEED value itself, however, has no bearing on where in a numerical sequence the NOISE pattern will begin. The numbers may fall in any sequence, so finding a desirable pattern of NOISE points is a trial and error process. You can experiment with over 2 billion possible NOISE patterns. Upon finding a useful NOISE pattern, whenever the same NSEED value is recalled, the same NOISE pattern will be reproduced.

**OFST (key)**      Type: MI Modifier

**Function:** Used to adjust the voltage offset of the waveform.

**Syntax:** OFST = 2

**Note:** See HIGH

**OP0**              Type: Command

**Function:** Enters Scope Draw mode.

**Syntax:** OP0

**Note:** ECHO must be enabled to see Time and Voltage on the display.

**OP4**              Type: Command

**Function:** Displays a menu for setting various general operating parameters.

**Syntax:** OP4

**Note:** ECHO must be enabled to see the menu on the display.

**OP7**              Type: Command

**Function:** Invokes OP7 mode. Similar but not identical to entering OP7 at front panel.

**Syntax:** OP7

**Note:** Entering OP7 at the front panel will show a list of special features on the display, but when sent remotely, ECHO must be enabled for the display to appear. Otherwise, OP7 mode will be active but the display will be blank and the RESTORE key must be pressed or the RES command sent to resume other functions.

**OP232**          Type: Command

**Function:** Used to configure RS-232 (serial) I/O Modifier parameters.

**Syntax:** OP232

**Note:** ECHO must be enabled if you wish to see the RS232 menu on the display.

**OP488**      Type: Command

Function: Used to configure IEEE-488 (parallel) I/O Modifier parameters.

Syntax: OP488

Note: ECHO must be enabled to see the IEEE-488 menu on the display.

**OP1510**      Type: Command

Function: Enters Tablet Draw mode.

Syntax: OP1510

Note: ECHO must be enabled if you wish to see the Tablet Draw opening message on the display.

**OP 6000**      Type: Command

Function: Used to retrieve captured waveforms from a D6000 series waveform analyzer.

Syntax: OP 6000

**OP 6001**      Type: Command

Function: Used to change the endpoint of captured waveforms already retrieved from a D6000 with OP 6000.

Syntax: OP 6001

**OP 9999**      Type: Command

Function: Deletes all saved files from non-volatile memory.

Syntax: OP 9999

Note: When sent remotely, OP 9999 will immediately execute the comprehensive deletion and will not perform a courtesy check as it does when entered at the front panel.

**PER**      Type: MD Modifier

**Function:** Sets the period length (in seconds) of the output waveform.

**Syntax:** PER = 32m

**Note:** If ECHO is enabled, the display will change the FREQ field to the PER format. This will apply to all the standard functions at the same time.

**PHS**      Type: MD Modifier

**Function:** Sets phase angle between SYNC OUT leading edge and first positive zero crossing of sine wave.

**Syntax:** PHS = +/- 0.99999, where the value is a fractional portion of one cycle

**Note:** As an example, PHS = 0.33333 yields one third of a cycle in phase shift, which is 120 degrees.

**PI (key)**      Type: Math Operator

**Function:** Used to represent the constant value 3.1415927 in a polynomial expression. Also -PI, representing the negative value.

**Syntax:** FOR 10m SIN(5K\*t)/((2\*PI)\*(5K\*t))

**Note:** PI is the 2nd shifted 0 key, labeled with the Greek letter. PI and -PI can also be lower case, pi and -pi.

**PLSW**      Type: MD Modifier

**Function:** Used to set the pulse width of a standard function square wave, in seconds.

**Syntax:** PLSW = 5u

**Note:** The SYM parameter for SINE mode will exhibit a corresponding change in the standard function menus.

**POLY**      Type: Command

**Function:** Puts machine in polynomial mode from standard function or MODIFIED modes.

**Syntax:** POLY

**PROMPT**      Type: MI Modifier

**Function:** When using the RS-232 option for communications, enables or disables the prompt (>) appearing on the remote controller screen. Any prompt appearing when using the GPIB (IEEE-488) interface must be generated by the controller.

**Syntax:** PROMPT = 0 (to disable), = 1 (to enable, default)

**PURGE**      Type: Command

**Function:** Deletes all waveforms from output memory.

**Syntax:** PURGE

**Note:** No courtesy check is performed.

**RAD (key)**      Type: Command

**Function:** Determines trigonometric arguments will be interpreted as radians per second.

**Syntax:** RAD (alternate setting of CYC/RAD key)

**Note:** See note under COS.

**RCL (key)**      Type: Command

**Function:** Recalls named waveform from memory.

**Syntax:** RCL FSIN

**REM (key)**      Type: Command

**Function:** Invokes remote mode, disabling all front panel keys except for the REM/LOC key and reset key pair (division & FORMAT).

**Syntax:** REM (alternate setting of REM/LOC key)

**Note:** When in remote mode, the REM indicator will light.

**RES**           Type: Command  
also RESTORE

**Function:** Exits OP mode or MODIFIED mode, suspends computation.

**Syntax:** RES

**Note:** Although this command emulates most RESTORE key functions, it does not duplicate the actual key press. RES will not enter MODIFIED mode as the key will. For that action, see the MOD command.

**RESET**       Type: Command  
also RSET

**Function:** Resets the machine, simulating power off then on.

**Syntax:** RESET or RSET

**RPT (key)**     Type: Math Operator

**Function:** Used to repeat waveform segments and/or provide n-shot capability where  $1 \leq n \leq 65535$ .

**Syntax:** RPT 1 (FOR 1m SIN(1K\*t))

**RUN (key)**     Type: Command

**Function:** Enables current output.

**Syntax:** RUN

**Note:** Key is labeled R/S (Run/Stop). When running, RUN indicator will light.

**SGN (key)**     Type: Math Operator

**Function:** This is the mathematical SIGNUM function, where the result of an evaluation or computation to which SGN is applied will be interpreted as a 1 if the result is positive, as a 0 if the result equals zero, or as a  $-1$  if the result is negative.

**Syntax:** AT 1m SGN(COS(1K\*t))

**Note:** Because the above example uses a COSINE function, the result of the computation will go from positive to negative. With the application of the SGN function, the waveform that will appear at the output will be a square wave with frequency and amplitude identical to the COSINE waveform defined in the polynomial.

**SIN (key)**    **Type:** Math Operator

**Function:** Defines the function in a polynomial which evaluates the sine of an expression.

**Syntax:** SIN(1K\*t)

**Note:** See note under COS.

**SNOI (key)**    **Type:** Command

**Function:** Duplicates pressing the NOISE standard function mode key, enables the generation of psuedo-random noise only.

**Syntax:** SNOI

**Note:** Bandwidth and amplitude of noise are selected with NBW and NAMP. If already in SNOI mode, sending SNOI will exit the noise function and enter POLY mode.

**SSIN (key)**    **Type:** Command

**Function:** Activates standard sine wave function.

**Syntax:** SSIN

**Note:** Key is labeled with sine wave symbol. If already in SSIN mode, sending SSIN will exit SSIN mode and enter POLY mode.

**SSQR (key)**    **Type:** Command

**Function:** Activates standard square wave function.

**Syntax:** SSQR

**Note:** Key is labeled with square wave symbol. If in SSQR mode, sending SSQR will exit SSQR mode and enter POLY mode.



**STOP (key)**      Type: Command

Function: Disables current output.

Syntax: STOP

Note: Key is labeled R/S.

**STRI (key)**      Type: Command

Function: Activates standard triangle wave function.

Syntax: STRI

Note: Key is labeled with triangle wave symbol. If in STRI mode, sending STRI will exit STRI mode and enter POLY mode.

**SYM**            Type: MD Modifier

Function: Used to set the symmetry of a sine or triangle waveform, in percent.

Syntax: SYM = 50 (default, balanced symmetry)

Note: Sending this command will also change the DUTY or PLSW parameter in SSQR function.

**T (key)**      Type: Math Operator

Function: Total (global) output time variable in equation, starting from the beginning of the waveform.

Syntax: FOR 1m SIN(1K\*t) FOR 2m SIN(1K\*T), where  $1m \leq T \leq 3m$

**t (key)**      Type: Math Operator

Function: Local time variable in equation segment, starting from the beginning of the relevant statement only.

Syntax: SIN(1K\*t) Initial value = 0

**TALKSRQ**      Type: MI Modifier

**Function:** Enables or disables the Talk SRQ, or output service request, which indicates readiness to transmit data.

**Syntax:** TALKSRQ = 0 (off, default) or = 1 (on)

**TAN (key)**      Type: Math Operator

**Function:** Defines the function in a polynomial which evaluates the tangent of the expression.

**Syntax:** FOR 1m TAN(1K\*t)

**Note:** See note under COS.

**TGTPNTS**      Type: MI Operator

**Function:** Abbreviation for *target points*, sets the approximate number of points to be computed.

**Syntax:** TGTPNTS = 750

**Note:** The number of points computed will be as close as possible to the target. The exact target number may or may not be realizable, depending on a variety of factors such as duration of a math function in relation to a clock rate. If forced by the demands of the parameters involved, the actual number of points computed can sometimes be many orders of magnitude different from the TGTPNTS value.

**TO (key)**      Type: Math Operator

**Function:** Specifies waveform segment end time. Referenced to global time.

**Syntax:** TO 10m 2.5 (means 2.5V output until T = 10m), TO 1m SIN(1K\*t)

**TRIG (key)**      Type: Command

also +TRIG

and -TRIG

**Function:** Specifies trigger mode in polynomial equations.

**Syntax:** FOR -TRIG, AT -TRIG, TO +TRIG, AT TRIG TO TRIG

**Note:** Refer to section E for a list of poly mode trigger syntax and the corresponding trigger modes.

**TRIGLEV**      Type: MI Modifier

**Function:** Used to set trigger level at one of two levels.

**Syntax:** TRIGLEV = 0 (sets level at 0V), = 1 (sets level at 1.5V, default)

**TRIGM**          Type: MD Modifier

**Function:** Used to select one of twelve triggering modes.

**Syntax:** TRIGM = 0 (default) through 11

**Note:** Triggering modes are assigned and perform as follows...

- = 0, FREE RUN : Output starts when computation is finished, free of any trigger requirement.
- = 1, +START : Output starts when next positive trigger edge occurs.
- = 2, –START : Output starts when next negative trigger edge occurs.
- = 3, +GATED : Output starts with positive trigger edges, stops at negative edges, restarts with positive edges.
- = 4, –GATED : Output starts with negative trigger edges, stops at positive edges, restarts with negative edges.
- = 5, +STOP : Output stops when next positive trigger edge occurs.
- = 6, –STOP : Output stops when next negative trigger edge occurs.
- = 7, +TO+ : Output toggles from start to stop or vice-versa whenever a positive trigger edge occurs.
- = 8, –TO– : Output toggles from start to stop or vice-versa whenever a negative trigger edge occurs.
- = 9, START : A GET will start output.
- = 10, STOP : A GET will stop output.

**Note:** Modes 9 and 10 are used only when GET (Group Execute Trigger) response is used as the trigger source in the OP488 configuration setup.

## SCOPE DRAW (OP 0)

### Description

---

Scope Draw mode enables you to modify the data which makes up a waveform while the waveform is displayed on a scope. Within the constraints of the full scale range and time period, you can modify single data points, draw ramps of any slope and length and draw DC levels of any amplitude and duration. The ramps and levels are generated by the resident AT and TO functions respectively. The cursor keys and the MARK key let you set single points and define the start and end points of the ramps and levels.

### Setup

---

Scope Draw can be used with the MARKER OUT signal to highlight a 32 clock period portion on the displayed waveform. MARKER OUT is a TTL pulse, and can be connected to the scope's Z-AXIS input. Depending on the configuration of the scope's Z-AXIS input, you may see a brightened or darkened area representing the marker pulse. This pulse can be used to highlight the start or end area of the portion of interest. You may have to adjust scope intensity along with a potentiometer inserted between MARKER OUT and Z AXIS IN on the scope to achieve the desired relative brightness levels of the whole waveform compared to the portion of interest.

To connect a scope for use with Scope Draw, a typical arrangement could be as follows:

Connect 2045 Output	To Scope Input
Front Panel, A or B OUT	CH 1
Rear Panel, SYNC OUT	External Trigger
Rear Panel, MARKER OUT	Z-Axis (Brightness)

### Entering and Using Scope Draw mode

---

To enter scope draw mode, you must first have a waveform in output memory. This can be any standard function (SIN, TRI or SQR, but not NOISE) or a polynomial entered from the keypad or recalled from the directory.

With this condition satisfied, press 2nd (blue shift key), OP, 0, ENTER. The display will show:

T=0.00000  
V=0.00000\*

\* or whatever level exists at T=0

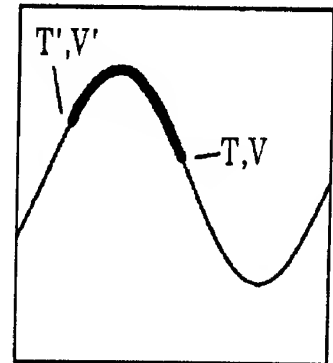
Upon entry into this mode, the output data residing in output memory is no longer classified as a standard function or a polynomial. Instead, the 2045 treats this data just like downloaded data which can now be modified. Initially, time T (current time) will always equal zero and voltage V (current voltage) will equal the output voltage at T=0. The Marker (used to highlight current time and voltage) also is set to zero time. To change the current time T, the left and right arrow cursor keys may be used. The time will change in units of 1 data point or about 1/TARGET PTS % of the period. As the time changes, the voltage will change according to the current voltage level. The highlighted marker will also change with the current time. A single press of the right arrow key will increase the time equivalent to one step (one point of output memory). The left arrow key will decrease the time in the same way. By holding the FAST key while pressing the left or right arrow key, you will change the time by a factor of 1/20th of the total time of the output waveform.

To modify a single point of data in the output waveform, step to the desired time and use the up and down arrow keys to adjust the output voltage. The level will change in units of about 0.0153% of FSR. Using the FAST key with an up or down key will enlarge the units of change to about 1.53% of FSR. To modify a range of points in a single segment, press the MARK key, which will bring up two more fields to the right of T and V:

T = 0.00000	T' = 0.00000
V = 0.00000	V' = 0.00000

T' and V' show the time and voltage where a reference point has been tagged with the MARK key. This tag is not the same as the MARKER OUT pulse, which is 32 clock periods long and can only be resolved to steps of every 32 clocks. The T' and V' coordinates define the start or the end point of a modified segment. To begin with, we'll call this the start point. Now use the arrow keys to move to some desired subsequent position. T and V now define the end point. Press the AT or the TO key, and the segment will become an interpolated ramp or a DC level from the start point to the end point or the time of the end point. The start point will always be the point with the lower time value, meaning the modification will always go from left to right. AT will connect two points at different times and levels. TO will draw a DC level from the point with the lowest time value to the time value at which the subsequent point occurs and not to the subsequent point itself unless its level is the same.

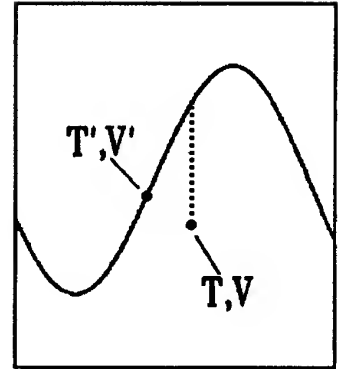
To illustrate some of this, look at the accompanying figures. In the first figure, T', V' and T, V define a segment on the waveform to be modified. This segment can be extended or shortened, following the track of the waveform, with the left or right cursor keys. The segment defines the start, end, and duration of the modification.



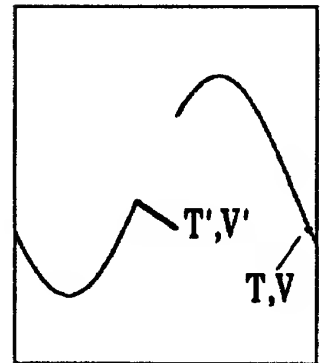
If the modification is an "AT" ramp, the second figure shows the result.



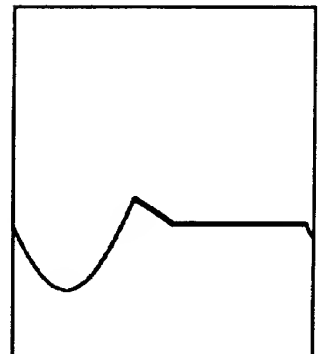
In the third figure, the end point has been pulled away from the waveform with the down cursor key so the modification can end off of the original waveform.



The result of an AT interpolation leaves the modified segment's end dangling out wherever you like as shown in the fourth figure. If you tag this end with the MARK key and advance T to somewhere near the edge of the screen...



...pressing the TO key will yield the result shown in the fifth figure.



To exit, press RESTORE. This will return the 2045 to polynomial mode regardless of whether or not you entered from polynomial mode.

The number, slope and/or duration of the AT and TO modifications are entirely up to you. And if you wish to modify your modifications, this can be done just as easily. Scope draw can be used in conjunction with tablet draw (OP 1510) and upload/download, making it possible to switch back and forth between these options, using the capabilities of each to operate on a waveform.



# TABLET DRAW (OP 1510)

## Description

---

Tablet Draw mode enables you to draw on the tablet any desired waveform shape and have it appear on a scope as you are drawing it. This allows for the quick generation of a shape which might otherwise be difficult and time consuming to translate into a polynomial.

Tablet Draw utilizes the Cherry model 1510 Graphics Tablet with a four key drawing puck.

## Setup

---

Note: Refer to the Cherry tablet user's manual for information not covered here.

1. Connect the RS-232 D type connector from the Cherry tablet to the RS-232 DTE port on the rear panel of the 2045. The DTE port is SERIAL 1, but this should not be selected as the command device, because the tablet is only a peripheral output device. To determine the current command device, enter OP 4 or check the rear panel dip switches. If the switches show 2 up and 3 down, this indicates SER1 is selected as the command device and this should be changed.
2. Connect the drawing puck to the tablet.
3. Set the blue jumper on the bottom of the tablet to the proper position for the drawing puck (see Cherry manual).
4. There are 16 option select switches on the bottom of the tablet, in two groups of eight. Two switch pairs (2-3; 7-8) and two switch triplets (9-10-11; 13-14-15) operate as groups on certain parameters. See the Cherry manual for more information. Set the switches to the following positions, counting from the left:

<u>Switch</u>	<u>Desired Parameter</u>	<u>Switch</u>	<u>Desired Parameter</u>
1-ON	Extended working area, no menu	9-OFF	Packed binary 2
2-ON	Switch stream mode	10-OFF	
3-OFF		11-ON	
4-OFF	Serial output	12-ON	No Line Feed required
5-ON	Green key on puck as Fn key	13-ON	9600 Baud
6-OFF	8 data bits	14-OFF	
7-ON	No parity	15-ON	
8-OFF		16-OFF	Not used

5. Connect the power adaptor to the tablet and plug the power cable into a standard outlet. This serves as the power switch.

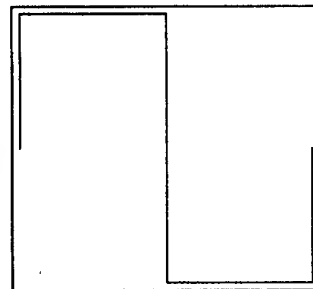


6. Connect the SYNC OUT on the rear panel of the 2045 to the external trigger input on a scope. The purpose of this is give the scope something consistent to trigger on while a waveform is being drawn or modified, so you will have a stable visible image.
7. Connect the FUNCTION OUT of the 2045 to the signal input of the scope.

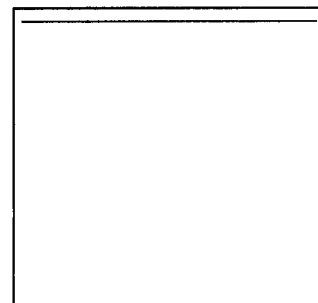
## Entering and Using Tablet Draw mode

When tablet draw mode is activated, the current RS-232 attributes (communication parameters) of 'SER1:' are automatically saved and then set to those of the Cherry tablet, ie: BAUD RATE = 9600, STOP BITS = 2, DUPLEX = HALF. Upon exiting the OP1510 mode, however, the 'SER1:' attributes are re-set to their original values.

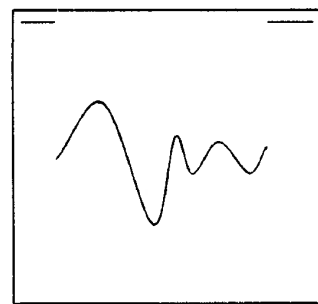
To enter tablet draw mode, you should first generate a waveform. This can be any standard function (SIN, TRI or SQR, but not NOISE) or a polynomial entered from the keypad or recalled from the directory. The 2045 need not be running (i.e. R/S light not lit) to enter draw mode, but something MUST be in output memory (i.e. xxx# points need to have been computed). The waveform to be edited must have been computed with 1000 or fewer data points: TARGET PTS cannot be greater than 1000. Since the resolution of Tablet Draw is 1000 data points, TARGET PTS must be set as close to 1000 as possible (see OP 4). If you start with a standard function square wave, you can make full use of the scope's screen when drawing by setting the scope to display only one full cycle of the square wave, taking up the whole screen horizontally and vertically as shown.



Press 2nd (blue shift key), OP, 1, 5, 1, 0, ENTER. An introductory message will appear on the 2045 display. The 2045 output data is unchanged at this point, but the output is no longer classified as a standard function or polynomial data set by the 2045. Instead, the data is treated just like downloaded data, which can now be edited or modified. You can draw on (modify) an existing wave shape, or a single press of the blue key on the drawing puck will place all the data points on a horizontal line at full scale maximum. This line displays all the points available for drawing. This allows you to create a new shape from the beginning. As you draw a waveform, the line will gradually disappear as the points are used in the waveform.



The wave you draw will appear on the scope just as you draw it, so if you clear the display and then start and stop drawing well inside the edges of the tablet, the unused data points will stay on the top line until the automatic fill is activated. The automatic fill will linearly interpolate the remaining data points into a line which will fill any gaps.



While drawing, the puck must be moved very slowly if you wish to pick up all or most of the points as you draw. The faster you move the puck, the fewer the number of points that will be picked up in the wave. Not picking up all the points while drawing is not necessarily a problem, because the automatic fill will most likely complete the job to your satisfaction. See the description below for the Red key.

The drawing puck can be used according to the functions assigned to the four colored keys:

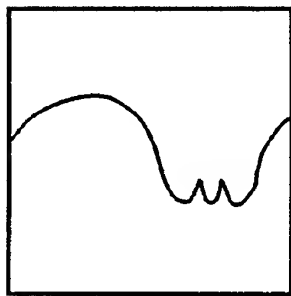
**Green key** - Pressing and holding this key while moving the puck will draw a wave. Where the drawing will appear is determined by where the puck is positioned on the tablet. You should move the puck slowly for better point pick up.

**Red key** - Pressing this key after drawing will fill in any points that were missed while drawing. The filling in is done with a linear interpolation utilizing Bresenham's Algorithm between existing points in the drawn wave. This feature also operates between the first point and the last point of the segment you have drawn, so any gap at the ends will be filled in. This automatically ensures a continuous periodic waveform.

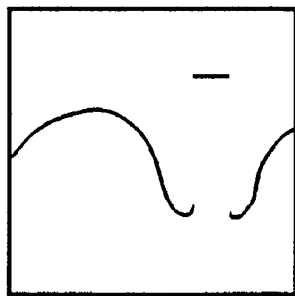
**Yellow key** - Pressing and holding this key while moving the puck horizontally will erase points from the drawn wave and put them back on the top line, behaving otherwise just like the green key.

**Blue key** - Pressing this key will erase the entire drawn wave, returning all points to the top line after which you can start drawing again.

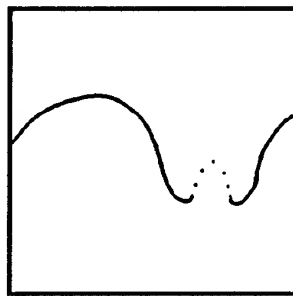
If you don't like a portion of the wave you have drawn, such as the double spike shown below (A), an easy way to fix this is to use the yellow key to erase the offending part (B) and then use the green key to place a few single points in a path outlining the preferred shape (C). Now if you press the red key, the erased points will be linearly interpolated between the single points, filling in the preferred shape (D).



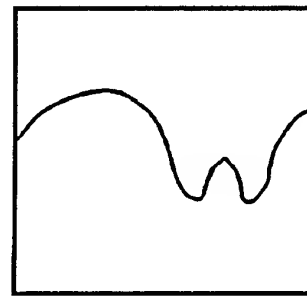
A



B



C



D

To disable the puck and exit tablet draw mode, press RESTORE. The 2045 will exit into polynomial mode whether or not you entered from poly mode.

You may want to scale the amplitude and/or time duration of the waveform you have created. If you press the RESTORE key twice (once to exit draw mode, again to enter MODIFIED mode), you can set the clock rate, amplitude, offset, marker location and trigger mode. The default values for these parameters are determined by the mode from which you entered tablet draw. The waveform time duration will be equal to the specified clock rate times the number of points computed. Note that you *cannot* utilize the noise parameters in Tablet draw.

Tablet draw can be used in conjunction with scope draw (OP 0) and upload/download, making it possible to switch back and forth between these options, using the capabilities of each to operate on a waveform.



# UPLOAD/DOWNLOAD

## I. Overview

---

Waveform data points may be uploaded from the 20XX's output memory to a host computer over either GPIB (IEEE-488) or RS-232 remote communications. Waveforms stored on the host in this way may then be downloaded back to the 20XX for subsequent output. The major advantage of this method over that of storing waveform equations is that the waveforms exist as sets of valid output points that can be recalled without recomputation. This is useful in avoiding the long computation time of a complex waveform expression, as well as providing a method of saving and recalling waveforms modified with SCOPE DRAW or TABLET DRAW (OP 0 and OP 1510, respectively).

## II. Implementation

---

There are four commands specific to Upload/Download procedures:

1. **UPLOAD** - to transfer waveform data currently in the 20XX output memory to an external device, such as a host computer.
2. **DNLOAD [filename]** - to transfer a data file from an external device back to the 20XX.
3. **CONT** - tells the 20XX to send the next data block (CONTinue sending data). Used only when UPLOADing over RS-232, not used with GPIB.
4. **BLOCK = [number of bytes]** - defines the maximum size of the 20XX data transfer during UPLOAD. **BLOCK = 0** causes the entire data file to be handled as a single block (See III. Upload-Block Size). Maximum size of a block can be 1000 bytes.

The UPLOAD and DNLOAD functions are in many ways similar to DATA download in the remote data mode. If a name is appended to the DNLOAD command, e.g. "DNLOAD SIN1" the new waveform will be added to the current waveforms in output memory and the entry "SIN1." will appear in the directory. Notice that the 20XX will append the name of a data file with '.' (a period). If no name is appended to the DNLOAD command, i.e. only "DNLOAD" is sent, the output file will simply be named "DATA." and all previously DNLOADed files (i.e. all directory files ending with '.') will be purged from output memory and deleted from the directory. Please note that the execution of any standard function or polynomial will also have this potentially undesirable effect of purging all existing DNLOADed waveforms from output memory.

If a filename is specified for the DOS file to be transferred to the 20XX and a further argument is given, for example "DN SIN1 ABCD" the filename "ABCD" will be sent with the string "DNLOAD", actually causing the waveform to be appended to those already in output memory and the entry "ABCD." to be added to the directory.

Waveforms will be transferred in three logical blocks:

1. A 194 byte waveform header block (contains total waveform size in bytes)
2. A 44 byte output memory attributes block
3. Waveform data block

The waveform header block and memory attributes block contain internal status information for the 20XX. These blocks are set by the 20XX to reflect its hardware configuration and **MUST NOT** be altered by the user. The only information that can be obtained from these blocks is the size of the waveform data file in bytes.

An error will be reported on the 20XX's screen if insufficient space is available in its output memory, if the DNLOAD target filename is invalid, or if an attempt is made to download a waveform that was originally computed on a 20XX that is different in any of three ways:

1. Memory configuration
2. Software revision level (i.e. 3.00 vs 4.00)
3. Maximum clock speed (i.e. 25 MHz vs 100 MHz)

### III. Upload-Block Size\*

---

Setting the maximum transfer block size on UPLOADs (via the command **BLOCK = xxxx**) is an important tool in tailoring the 20XX's I/O to the specific limitations/capabilities of your host computer.

The waveform data transmitted during an **UPLOAD** is not in standard ASCII format. Rather, it is **BINARY** or **UNFORMATTED**. This means that bytes that normally would be recognized as control characters (ie. carriage return, line feed) may be embedded in the data and **MUST NOT** be interpreted as such. All reads must therefore be **BINARY** or **UNFORMATTED**, ie. all reads must terminate upon reaching a specified requested read-count, *not* upon receiving a carriage return, for example. It is for this reason that the lengths of all data transmissions have been made available.

Setting the block size to 0 will cause each **UPLOAD** waveform data 'block' (ie. header, attributes or DATA) to be transmitted as one contiguous block. This is needed when performing a DMA transfer. A user that does not have DMA capabilities must send a "**BLOCK = xxxx**" command to cause the waveform data to be transferred in smaller, more manageable blocks. The block size "**xxxx**", which must be set to an integer value between 1 and 1000, inclusive., represents the maximum number of bytes that the 20XX will transfer at one time in response to a read from the host.

Since all reads are **BINARY** or **UNFORMATTED**, the user's program must specify the exact number of bytes for each read. The size of the **HEADER** and **ATTRIBUTES** blocks are known, leaving only the size of the actual waveform data undetermined. This information, however, is contained in the **HEADER**:

The PC should look at bytes 6-13 of the **HEADER** to get the actual number of data bytes.

byte:	0	1	2	3	4	5	6	7	8	9	10	11	12	13
value:	34	2E	30	30	00	00	00	00	00	18	00	00	07	C0
meaning:	↑	rev. level			↑		↑	datalen1			↑	datalen2		↑

Example:

```

rev level = "4.00"
datalen1 = |00 |00 |00 |18 |
datalen2 = |00 |00 |07 |C0 |

```

total number of bytes in data block = datalen1 + datalen2 = 00 00 07 D8 (hex)

*\*Note:* The Block size does not affect data transfers in "DNLOAD" mode.

### a. RS232 Uploads

Example: If a PC is limited to reading only 80 characters (bytes) at a time, then send the command:

"BLOCK = 80"

before sending

"UPLOAD"

This will cause the 20XX to send no more than 80 characters at a time.

A sample upload would proceed as follows:

```

20XX:  sends first 80 bytes of header
PC:    "CONT" (continue)
20XX:  sends bytes 81-160 of header
PC:    "CONT"
20XX:  sends bytes 161-194 of header
PC:    "CONT"
20XX:  sends 44 bytes of attributes
PC:    "CONT"
RPT:   20XX:  sends up to 80 bytes of data
PC:    "CONT"
GOTO RPT until all data has been sent.

```

*Note:* Sending anything other than "CONT" from PC will cause upload to abort.

### b. GPIB Uploads

It is highly recommended that the command "EOI = 1" be sent prior to "UPLOAD". This will enable the host to terminate a read when an EOI is received, even if the requested read-count was too long. For purposes of illustration, examples below include EOI set to both ON and OFF.

*Note:* Do not send "CONT" when UPLOADing across GPIB.

#### GPIB - UPLOAD to host with DMA capabilities

*Note:* Typically send "BLOCK = 0", "EOI = 1" for DMA transfers.

#### Example :

BLOCK = 0, EOI = ON

send header (one continuous block with <EOI>)

send attributes (one continuous block with <EOI>)

send DATA (one continuous block with <EOI>)

#### Example :

BLOCK = 0, EOI = OFF

send header (one continuous block without <EOI>)

send attributes (one continuous block without <EOI>)

send DATA (one continuous block without <EOI>)

*GPIB - UPLOAD to host without DMA capabilities*

Send "BLOCK = xxxx", where xxxx represents the maximum number of bytes for each block.

NOTE: Internally, data transfers on 20XX are limited to 1000 bytes.

**Example :**

BLOCK = 1000, EOI = ON

send header (one or more blocks  $\leq 1000$ , each with <EOI>)

send attributes (one or more blocks  $\leq 1000$ , each with <EOI>)

send DATA (one or more blocks  $\leq 1000$ , each with <EOI>)

**Example :**

BLOCK = 1000, EOI = OFF

send header (one or more blocks  $\leq 1000$ , each without <EOI>)

send attributes (one or more blocks  $\leq 1000$ , each without <EOI>)

send DATA (one or more blocks  $\leq 1000$ , each without <EOI>)

Set EOI to ON or send "EOI = 1", so the 20XX will set EOI with the end of every transfer.

Note: BLOCK > 0 means EOI at end of every block.

## IV. GPIB Command Program and Application Note

---

A program written in BASIC concludes this section. This program is a suggested format for automating the upload/download process.

### APPLICATION NOTE:

Each block transmitted from the 20XX is terminated with EOI set. When performing a DMA transfer, using the EOI as a terminator forces the application program to request the exact size of the block sent or suffer one of two consequences:

1. If too many characters are sent, the program will read only as many as it requested and then drop the handshake, causing the GPIB driver in the 20XX to 'lock up'. This could be avoided if the program immediately went back for another read, but the act of addressing the 20XX to 'talk' again causes the loss of one byte of data. Put another way, if the program asynchronously takes control of the bus to request more data while the 20XX is still transmitting, the data byte currently in the PC488 (on the bus) is lost.
2. If too few characters are sent, the GPIB drivers in the application program (as provided by CEC) will return with the data and actual count only after a "time out". This can be very time consuming (not to mention frustrating!). The only reasonable solution, other than re-writing the low-level GPIB drivers provided with the board, is to request DMA transfer only once (i.e. address the 20XX to talk ONE TIME after sending it the "UPLOAD" string) and ignore all subsequent EOIs.

You may note that this method will not work unless the application program knows the total size of the transmission to follow. Fortunately, this information can be extracted from the second and third long-word of the 194 byte header. Because the number of bytes required for a waveform is contained in the 194 byte header, the application program could terminate I/O on byte-count and not on EOI (i.e. it just keeps going back for more until there isn't any left). The PC's DMA read/writes are therefore independent of the I/O block size requirements of the 20XX.

Due to hardware limitations, the waveform data points cannot be loaded directly into Output Memory via the DMA chip on the 20XX's CPU card. The DMA controller can only transfer the points to a RAM buffer. The CPU then transfers the RAM buffer contents to the Output Memory. The size of this RAM buffer is currently 1280 bytes.

*Note:* Data is transmitted in unformatted or binary mode, not ASCII. The software must be able to do binary read/write. Transfers are terminated on byte count, unlike ASCII which terminates on control characters.



```

*****
'*      GPIB Command program to UPLOAD data from/DNLOAD data to      *
'*      the DATA 2000/2020/2040/2045 series                          *
'*      Polynomial Waveform Synthesizers.                             *
'*                                                                      *
'*      This program was developed under MICROSOFT QuickBASIC 3.00,   *
'*      using the Capital Equipment Corp.'s PC<>488 GPIB interface card. *
'*                                                                      *
'*      WCW  Feb 15, 1989    ANALOGIC, Data Precision division        *
'*-----*
'*
'*      NOTE:  This program is intended to illustrate both the protocol *
'*      for UPLOAD/DNLOAD as well as the use of the Serial Poll Response *
'*      word in determining the Data 20xx's state.                      *
'*
'*      In order to show this, the TALK, LISTEN and ERROR SRQs are     *
'*      enabled on the 20xx. All reads and writes are performed within  *
'*      a polling loop in which the PC first Serial Polls the 20xx and  *
'*      then waits for the appropriate Serial Poll Response word.       *
'*
'*      The PC-to-Data 20xx communications can be implemented in one   *
'*      of three ways:
'*
'*      1.)  Enable all SRQs and provide the PC with the capability    *
'*            of servicing the GPIB board's SRQ interrupts. This method *
'*            requires the writing of an SRQ interrupt service routine    *
'*            to Serial Poll the 20xx (thereby clearing the SRQ) and     *
'*            act upon the returned status byte.
'*
'*      2.)  Disable all SRQs and implement Serial Poll Response       *
'*            status polling loop. This is the approach used in this    *
'*            program. It is interesting to note that the SRQ's can     *
'*            be left ON/enabled without causing the bus to 'lock-up'   *
'*            because the act of Serial Polling the 20xx will clear     *
'*            the pending SRQ.
'*
'*      3.)  Disable all SRQs and don't Serial Poll the device.       *
'*            This 'sledge-hammer' method will force you to either      *
'*            robust command procedures (similar to those required for  *
'*            UPLOAD/DNLOAD) or rely on 'time-outs' to deal with        *
'*            commands such as 'DIR' and 'DUMP'.
'*
*****

```

```

'
'
CLS
SCREEN 0
'
'   CEC PC<>488 routine offsets:
INIT%=0 : XMIT%=30 : RECV%=33 : SEND%=36 : ENTER%=39
TARRAY%=200 : RARRAY%=203 : DMA2%=206
SPOLL%=12 : PPOLL%=15
DMAIN%=&H2105 : DMAOUT%=&H2109           ' Bit codes for DMA read/write
'
D20XX%=16                               ' 2000 is at GPIB address 16
MY.ADDR%=21 : SYS.CONT%=0                ' PC (controller) is at address 21

EOIOUT%=0                               ' Do not send an EOI on output
REPORT%=0                               ' Do not report SPOLL status word to screen
SCRIPT%=0                               ' Do not save session to script file

OPTION BASE 0
DIM WORD!(10)

'***** NOTE:   Set the block size for all transmissions here ! *****'
BLOCK%=500                               ' INDIGIT%() must dimensioned to be '*'
DIM INDIGIT%(500)                        ' AT LEAST as big as BLOCK%          '*'
CLS
GOSUB Config20XX
END

Config20XX:
'*****
'*      Re-Configure the 20xx                      *
'*****

ResetSYS:
PRINT " ": PRINT "      Re-setting the GPIB interface . . . ."

DEF SEG = &HC000                          ' Initialize CEC GPIB card
CALL ABSOLUTE (MY.ADDR%,SYS.CONT%,INIT%)
'
OUTCHAR$="REN"                            ' Initialize bus with REMOTE ENABLE
CALL ABSOLUTE(OUTCHAR$,STATUS%,XMIT%)

' Set up 20xx with a default command interface configuration.
GOSUB SerialPoll
A$="PROMPT=0 ECHO=1 ERRSRQ=1 TALKSRQ=1 LISNSRQ=1"
GOSUB SendCMD

```

```

'
TryAgain:
'CLS
PRINT " "
PRINT " ***** "
PRINT "***  Program to transfer data between a Data 20XX and a PC ***"
PRINT " ***** "
PRINT " "
'   INPUT "   Enter Command Mode or Quit (C/Q):  ",Q$
'   IF (LEN(Q$)=0) THEN GOTO TryAgain
'   IF (Q$="Q")OR(Q$="q") THEN GOTO EndPgm
'   IF (Q$="C")OR(Q$="c") THEN GOTO SelCMD
'   GOTO TryAgain
'   SelCMD:
'       GOSUB CMDMode
'   IF STAT%<>0 THEN GOTO BadReturn
'       INPUT " Hit <enter> to continue:  ",Q$
'       GOTO TryAgain
BadReturn:
'   PRINT " Operation failed, must reset the GPIB interface bus."
'   GOTO ResetSYS
EndPgm:
RETURN

```

CMDMode:

```

'*****
'*      CMD Mode
'*      Enter command mode with the Data 20XX.
'*****
PRINT "  Data 20xx Command Interface Mode."
PRINT " ('EXIT' to quit, 'HELP' for help)"

```

GetCMD:

```

A$=SPACE$(80) : INPUT "> ",A$
IF A$ = "EXIT"   THEN GOTO DoneCMD
IF A$ <> "UPLOAD" THEN GOTO CHK1
  GOSUB UpLoad : GOTO GetCMD
CHK1:
IF A$ <> "DNLOAD" THEN GOTO CHK2
  GOSUB DnLoad : GOTO GetCMD
CHK2:
IF A$ <> "SPOLL" THEN GOTO CHK3
  GOSUB GetSPOLL : GOTO GetCMD
CHK3:
IF A$ <> "RESTART" THEN GOTO CHK4
  STAT%=-1 : RETURN
CHK4:
IF A$ <> "REPORT" THEN GOTO CHK5
  REPORT%=1 : GOTO GetCMD
CHK5:
IF A$ <> "NOREPORT" THEN GOTO CHK6
  REPORT%=0 : GOTO GetCMD
CHK6:
IF A$ <> "HELP" THEN GOTO CHK7
  GOSUB Help : GOTO GetCMD
CHK7:
IF A$ <> "SCRIPT" THEN GOTO CHK8
  GOSUB Script : GOTO GetCMD
CHK8:
IF A$ <> "NOSCRIPT" THEN GOTO CHK9
  GOSUB NoScript : GOTO GetCMD
CHK9:
IF A$ <> "BATCH" THEN GOTO CHK10
  GOSUB Batch : GOTO GetCMD
CHK10:
  GOSUB SendCMD
  GOTO GetCMD

```

DoneCMD:

```
STAT% = 0
```

RETURN

## Help:

```

CLS : PRINT "" : PRINT ""
PRINT "EXIT      Quit command session."
PRINT "RESTART   Re-initialize Data 20xx configuration."
PRINT ""
PRINT "SPOLL      Serial Poll the Data 20xx and enter status loop."
PRINT "REPORT      Enable reporting of Serial Poll Response word."
PRINT "NOREPORT    Disable reporting of Serial Poll Response word."
PRINT ""
PRINT "UPLOAD      Save Data 20xx waveform on PC."
PRINT "DNLOAD       Transfer PC waveform copy to Data 20xx."
PRINT ""
PRINT "SCRIPT       Open file SCRIPT.BAS to copy all I/O to."
PRINT "NOSCRIPT      Close file SCRIPT.BAS."
PRINT ""
RETURN

```

## Script:

```

OPEN "SCRIPT.BAS" FOR OUTPUT AS #2
SCRIPT%=1
RETURN

```

## NoScript:

```

CLOSE #2
SCRIPT%=0
RETURN

```

## Batch:

```

INPUT "Enter name of file containing commands: ", BATCH$
OPEN BATCH$ FOR INPUT AS #3
BatchMore:
INPUT #3, A$
IF LEN(A$)=0 THEN GOTO BatchDone
IF ((LEN(A$)=1) AND ((A$=CHR$(10))OR(A$=CHR$(13)))) THEN GOTO BatchDone
GOSUB SendCMD
GOTO BatchMore
BatchDone:
CLOSE #3
RETURN

```

UpLoad:

```

*****
'*      UPLOAD                                     *
'*      Perform an UPLOAD.                         *
*****
PRINT "  UPLOAD waveform data from a Data "+DEVICE$+" to a PC."
'
GOSUB GetConfig
OPEN FILE$ FOR RANDOM ACCESS WRITE AS #1 LEN=2
FIELD #1, 2 AS WORDSTR$ : REC1CNT!=1
'
A$="PROMPT=0 ECHO=1 ERRSRQ=0 TALKSRQ=1 LISNSRQ=1"
GOSUB SendCMD
OUTCHAR$="UPLOAD"      : GOSUB WriteGPIO
'
GOSUB SRQTalk : IF STAT%<>0 THEN RETURN
'
GOSUB AddrTALK          ' Address device to TALK
'
INLEN%=BLOCK%           ' Get the first block of the HEADER:
IF (INLEN% > HDRLEN!) THEN INLEN%=HDRLEN!
GOSUB ReadARRAY         '   Read data from 20XX
GOSUB UnAddr
GOSUB WriteDataToFile   '   Output to waveform file
GOSUB GetLen            '   Get DATALEN = MCLen + OMEMLEN
PRINT " from Data "+DEVICE$+" to ";FILE$;" . . . "
'
TOTALEN! =(HDRLEN!-BLOCK%) ' Get remaining bytes of HEADER
IF (TOTALEN!>0) THEN GOSUB GetDataFrom20XX
'
TOTALEN! = MEMATTR!      ' Get MEMORY ATTRIBUTES block
GOSUB GetDataFrom20XX
'
TOTALEN! = DATALEN!     ' Get waveform data
GOSUB GetDataFrom20XX
'
XferDone:
GOSUB UnAddr             ' Un-address device
PRINT " Transfer complete."
A$="RUN" : GOSUB SendCMD  ' Turn ON the 20XX's output
A$="LOC" : GOSUB SendCMD  ' Return the 20XX to local
CLOSE #1
STAT%=0
RETURN

```

DnLoad:

```

'*****
'*      DNLOAD                                          *
'*      Perform a DNLOAD.                              *
'*****
PRINT "  DNLOAD waveform data from a PC to a Data "+DEVICE$
'
GOSUB GetConfig
'
FILE20XX$ = ""
PRINT "Target file name on Data "+DEVICE$+" system"
INPUT " (default - 'DATA'):" ,FILE20XX$
'
OPEN FILE$ FOR RANDOM ACCESS READ AS #1 LEN=2
FIELD #1, 2 AS WORDSTR$ : REC1CNT!=1
'
OUTCHAR$="DNLOAD "+FILE20XX$      : GOSUB WriteGPB
'
' If the device is not ready to listen, then we must abort xfer
GOSUB SRQListen : IF STAT%<>0 THEN RETURN
GOSUB AddrLISTEN          ' Address device to LISTEN
'
INLEN%=BLOCK%              ' Get the first block of the HEADER:
IF (INLEN% > HDRLEN!) THEN INLEN%=HDRLEN!
GOSUB ReadDataFromFile      ' Read data from waveform file
GOSUB GetLen                ' Get DATALEN = MCLEN + OMEMLEN
PRINT " from ";FILE$;" to "+DEVICE$+" . . . "
GOSUB WriteArray            ' Send to 20XX
'
TOTALEN! =(HDRLEN!-BLOCK%)  ' Send remaining bytes of HEADER
IF (TOTALEN!>0) THEN GOSUB SendDataTo20XX
'
TOTALEN! = MEMATTR!         ' Send MEMORY ATTRIBUTES block to 20XX
GOSUB SendDataTo20XX
'
TOTALEN! = DATALEN!        ' Send waveform data to 20XX
GOSUB SendDataTo20XX
'
GOTO XferDone

```

GetConfig:

```

'*****
'*      GET CONFIG
'*      Get User's Requests.
'*****
PRINT " "
PRINT "Are the transfers to be between a PC and a 2020 or a 2040?"
INPUT "(answer '2020' or '2040') ",DEVICE$
IF DEVICE$="2040" THEN GOTO Config2040
IF DEVICE$="2020" THEN GOTO Config2020
GOTO Config20XX
Config2020:
    HDRLen!=194
    MEMATTR!=44
    GOTO GetDMA
Config2040:
    HDRLen! =262
    MEMATTR!=92
    ' 194 byte HEADER
    ' 44 byte MEMORY ATTRIBUTES
    ' 262 byte HEADER
    ' 92 byte MEMORY ATTRIBUTES
,
GetDMA:
    INPUT "Do you wish to use the CEC's DMA capabilities (Y/N) : ",Q$
    IF (LEN(Q$)=0) THEN Q$="N"
    Q$=MID$(Q$,1,1)
    IF (Q$="Y") OR (Q$="y") THEN MODE$="DMA" ELSE MODE$="ARRAY"
    ,
    PRINT "Do you wish to use the "+DEVICE$+"'s DMA capabilities";
    INPUT " (Y/N) : ",Q$
    IF (LEN(Q$)=0) THEN Q$="N"
    Q$=MID$(Q$,1,1)
    IF (Q$="Y") OR (Q$="y") THEN FASTIO$="FASTIO=1" ELSE FASTIO$="FASTIO=0"
    ,
    'Get PC target filename
    INPUT "DOS file name to xfer to/from (default - TMP.WAV): ",FILE$
    IF LEN(FILE$)=0 THEN FILE$="TMP.WAV"
    ,
    'Set the 20xx's configuration now.
    A$="BLOCK="+STR$(BLOCK%) : GOSUB SendCMD
    A$="ECHO=1" : GOSUB SendCMD
    A$= FASTIO$ : GOSUB SendCMD
RETURN

```



```

*****
*                               Send a command to the Data 20XX                               *
*****

```

SendCMD:

```

B$=SPACE$(80) : POLL%=0
CALL ABSOLUTE(D20XX%,POLL%,STATUS%,SPOLL%)
IF STATUS%<>0 THEN GOTO CommandSent
IF (POLL% AND 4)=0 THEN GOTO ReportSRQ
IF SCRIPT%=1 THEN PRINT #2, "> "+A$

```

SendERROR:

```

CALL ABSOLUTE(D20XX%,A$,STATUS%,SEND%)
FOR I = 1 TO 5000 : NEXT I

```

GetSPOLL:

```

B$=SPACE$(80) : POLL%= 0
CALL ABSOLUTE(D20XX%,POLL%,STATUS%,SPOLL%)
IF STATUS%<>0 THEN GOTO CommandSent

```

ReportSRQ:

```

IF REPORT% = 0 THEN GOTO ServiceSRQ
PRINT "spoll: "+STR$(POLL%);
IF (POLL% AND 64) THEN PRINT " S";
IF (POLL% AND 1) THEN PRINT " E";
IF (POLL% AND 2) THEN PRINT " T";
IF (POLL% AND 4) THEN PRINT " L";
PRINT ""

```

ServiceSRQ:

```

IF (POLL% = 0) THEN GOTO GetSPOLL
IF (POLL% AND 2) THEN GOTO TalkSRQ
IF (POLL% AND 1) THEN GOTO ErrorSRQ
IF (POLL% AND 4) THEN GOTO ListenSRQ
GOTO CommandSent

```

TalkSRQ:

```

B$=SPACE$(80) : STATUS%=0
CALL ABSOLUTE(B$,LENGTH%,D20XX%,STATUS%,ENTER%)
IF STATUS%<>0 THEN GOTO CommandSent
IF LEN(B$)=0 THEN GOTO CommandSent
IF ( (LEN(B$)=1) AND ((B$=CHR$(10))OR(B$=CHR$(13))) ) THEN GOTO Command
PRINT B$
IF SCRIPT%=1 THEN PRINT #2, B$
GOTO GetSPOLL

```

ErrorSRQ:

```

A$="ERROR"
GOTO SendERROR

```

ListenSRQ:

CommandSent:

```

RETURN

```

## SendDataTo20XX:

```

*****
'*                               Send data to the 20XX                               *
'*-----*
'* This procedure sends data, previously saved to a DOS file *
'* by an UPLOAD, to the Data 20XX during a DNLOAD operation. *
'*-----*
LOOPCNT!=FIX(TOTALN!/BLOCK%) : LASTCNT=TOTALN!-(LOOPCNT!*BLOCK%)
CNT%=0
IF LOOPCNT!=0 THEN GOTO SDRemain
FOR LP = 1 TO LOOPCNT!
  INLEN%=BLOCK% : GOSUB ReadDataFromFile ' Read in the data from file
  ' If the device is not ready to talk, then we must abort xfer
  GOSUB SRQListen : IF STAT%<>0 THEN RETURN
  GOSUB AddrLISTEN ' Address device to LISTEN
  GOSUB WriteARRAY : CNT%=CNT%+INLEN% ' Write the data to the 20XX
NEXT LP
SDRemain:
IF LASTCNT=0 THEN RETURN
INLEN%=LASTCNT : GOSUB ReadDataFromFile ' Read in the remainder
' If the device is not ready to listen, then we must abort xfer
GOSUB SRQListen : IF STAT%<>0 THEN RETURN
GOSUB AddrLISTEN ' Address device to LISTEN
GOSUB WriteARRAY : CNT%=CNT%+INLEN% ' Write the data to the 20XX
RETURN

```

## GetDataFrom20XX:

```

*****
'*                               Get data from the 20XX                               *
'*-----*
'* This procedure gets data from the Data 20XX and saves it to *
'* a DOS file during a DNLOAD operation. *
'*-----*
LOOPCNT!=FIX(TOTALN!/BLOCK%) : LASTCNT=TOTALN!-(LOOPCNT!*BLOCK%)
CNT%=0
IF LOOPCNT!=0 THEN GOTO GDRemain
FOR LP = 1 TO LOOPCNT!
  INLEN%=BLOCK%
  ' If the device is not ready to talk, then we must abort xfer
  GOSUB SRQTalk : IF STAT%<>0 THEN RETURN
  GOSUB AddrTALK ' Address device to TALK
  GOSUB ReadARRAY ' Read in the data
  GOSUB UnAddr
  GOSUB WriteDataToFile : CNT%=CNT%+INLEN% ' Write the data to a file
NEXT LP
GDRemain:
IF LASTCNT=0 THEN RETURN
INLEN%=LASTCNT
' If the device is not ready to talk, then we must abort xfer
GOSUB SRQTalk : IF STAT%<>0 THEN RETURN
GOSUB AddrTALK ' Address device to TALK
GOSUB ReadARRAY ' Read in the remainder
GOSUB UnAddr
GOSUB WriteDataToFile : CNT%=CNT%+INLEN% ' Write the data to a file
RETURN

```

## WriteDataToFile:

```

'*****
'*          Write 20XX UPLOAD data to a DOS file          *
'*****
  K%=(INLEN%/2) : L%=INLEN% - (K%*2)
  FOR J% = 0 TO (K%-1)
    RSET WORDSTR$ = MKI$(INDIGIT%(J%))
    PUT #1,REC1CNT! : REC1CNT!=REC1CNT!+1
  NEXT J%
  IF L%>0 THEN PRINT "WARNING:  An odd number of bytes read from 20XX."
RETURN

```

## ReadDataFromFile:

```

'*****
'*          Read 20XX DNLOAD data from a DOS file          *
'*****
  K%=(INLEN%/2) : L%=INLEN% - (K%*2)
  FOR J% = 0 TO (K%-1)
    GET #1,REC1CNT! : REC1CNT!=REC1CNT!+1
    INDIGIT%(J%)=CVI(WORDSTR$)
  NEXT J%
  IF L%>0 THEN PRINT "WARNING:  An odd number of bytes read from ";FILE$
RETURN

```

## GetLen:

```

'*****
'*  Calculate number length of wave data bytes in an UPLOAD or DNLOAD  *
'*****
  FOR J%=3 TO 6
    ' Swap the bytes of each word read in.
    LOW% = ( (INDIGIT%(J%) AND (&HFF00)) / &H100 ) AND (&H00FF)
    HIGH% = INDIGIT%(J%) AND (&H00FF)
    WORD!(J%) = HIGH% : WORD!(J%) = (WORD!(J%) * &H100) + (LOW%)
  NEXT J%
  MCLen! = (WORD!(3) * 65536!) + WORD!(4) ' Build the lengths
  OMeMlen! = (WORD!(5) * 65536!) + WORD!(6)
  DATALEN! = MCLen! + OMeMlen!
  PRINT " Transferring ";STR$(HDRLEN!+MEMATTR!);"bytes of Waveform header"
  PRINT " and";STR$(DATALEN!);" bytes of Waveform data ";
RETURN

```

```

*****
'*                               GPIB DRIVER ROUTINES                               *
*****

```

SerialPoll:

```

'-----'
'  Serial Poll a device.  '
'-----'
  B$=SPACE$(80) : POLL%= 0 : STAT%=0
  CALL ABSOLUTE(D20XX%,POLL%,STAT%,SPOLL%)
  IF POLL%=0 THEN GOTO SerialPoll
  RETURN

```

AddrTALK:

```

'-----'
'  Address device to TALK.  '
'-----'
  TEMPOUT$="UNL UNT TALK "+STR$(D20XX%)+ " MLA"
  STAT%=0 : CALL ABSOLUTE(TEMPOUT$,STAT%,XMIT%)
  IF STAT%=0 THEN RETURN
  PRINT "GPIB Error addressing device "+STR$(D20XX%)+ " to TALK."
  PRINT "STATUS: ";STAT%
  GOTO EndPgm

```

AddrLISTEN:

```

'-----'
'  Address device to LISTEN.  '
'-----'
  TEMPOUT$="UNL UNT REN LISTEN "+STR$(D20XX%)+ " MTA"
  STAT%=0 : CALL ABSOLUTE(TEMPOUT$,STAT%,XMIT%)
  IF STAT%=0 THEN RETURN
  PRINT "GPIB Error addressing device "+STR$(D20XX%)+ " to LISTEN."
  PRINT "STATUS: ";STAT%
  GOTO EndPgm

```

UnAddr:

```

'-----'
'  Un-Address device.  '
'-----'
  TEMPOUT$="UNT UNL"
  STAT%=0 : CALL ABSOLUTE(TEMPOUT$,STAT%,XMIT%)
  IF STAT%=0 THEN RETURN
  PRINT "GPIB Error Un-addressing devices, STATUS: ";STAT%
  GOTO EndPgm

```

SRQTalk:

```

-----
' Check for a TALK SRQ
-----
'FOR I = 1 TO 5000 : 'NEXT I
GOSUB SerialPoll : STAT%=-1
IF (POLL% AND 2) THEN STAT%=0
RETURN

```

SRQListen:

```

-----
' Check for a LISTEN SRQ
-----
'FOR I = 1 TO 5000 : 'NEXT I
GOSUB SerialPoll : STAT%=-1
IF (POLL% AND 4) THEN STAT%=0
RETURN

```

SRQError:

```

-----
' Check for a ERROR SRQ
-----
'FOR I = 1 TO 5000 : 'NEXT I
GOSUB SerialPoll : STAT%=0
IF (POLL% AND 1) THEN STAT%=-1
RETURN

```

ReadGPIB:

```

-----
' GPIB INPUT
' Reads formatted character strings.
-----

```

```

GOSUB SRQTalk
IF STAT%<>0 THEN GOTO RGnotok
GOSUB AddrTALK ' Address device to talk.
INCHAR$=SPACE$(INLEN%) : STAT%=0
CALL ABSOLUTE(INCHAR$,INLEN%,STAT%,RCV%)
IF STAT%=0 THEN GOTO RGok
RGnotok:
PRINT "GPIB RECEIVE Input Error" : GOTO EndPgm
RGok:
GOSUB UnAddr ' Un-Address device.
INCHAR$=MID$(INCHAR$,1,INLEN%)
RETURN

```

WriteGPIB:

```

-----
                GPIB OUTPUT
                Writes formatted character strings.
-----
PRINT " Sending command: '"+OUTCHAR$+"'"
GOSUB SRQListen
IF STAT%<>0 THEN GOTO WGnotok
GOSUB AddrLISTEN                ' Address device to listen.
TEMPOUT$="DATA '"+OUTCHAR$+"'" END" : STAT%=0
CALL ABSOLUTE(TEMPOUT$,STAT%,XMIT%)
IF STAT%=0 THEN GOTO WGok
WGnotok:
PRINT "GPIB SEND Output Error, STATUS:"STAT% : GOTO EndPgm
WGok:
GOSUB UnAddr                ' Un-Address device.
RETURN

```

ReadARRAY:

```

-----
                GPIB BINARY ARRAY INPUT
                Reads un-formated (binary) character arrays
-----
IF MODE$="DMA" THEN GOTO ReadDMA
OFS%=0 : STAT%=0 : SEGM%=-1 : STAT%=0
REQLEN%=INLEN% : INLEN%=0
OFS%=VARPTR(INDIGIT%(0))                ' Get addr of str descriptor,
CALL ABSOLUTE(SEGM%,OFS%,REQLEN%,INLEN%,STAT%,RARRAY%) 'Read bytes.
IF (STAT%=0)OR(STAT%=32) THEN RETURN
PRINT "GPIB RARRAY Input Error, STATUS:";STAT%
GOTO EndPgm

```

WriteARRAY:

```

-----
                GPIB BINARY ARRAY OUTPUT
                Writes un-formated (binary) character arrays
-----
IF MODE$="DMA" THEN GOTO WritedDMA
OFS%=0 : STAT%=0 : SEGM%=-1 : STAT%=0
REQLEN%=INLEN% : 'INLEN%=0
OFS%=VARPTR(INDIGIT%(0))                ' Get addr of str descriptor,
CALL ABSOLUTE(SEGM%,OFS%,REQLEN%,EOIOUT%,STAT%,TARRAY%) ' Read bytes.
IF STAT%=0 THEN RETURN
PRINT "GPIB TARRAY Output Error, STATUS:";STAT%
GOTO EndPgm

```

ReadDMA:

```

-----
DMA INPUT
Reads un-formated (binary) character arrays
-----
OFS%=0 : STAT%=0 : SEGM%=-1 : STAT%=0
REQLEN%=INLEN%
OFS%=VARPTR(INDIGIT%(0))          ' Get addr of str descriptor,
CALL ABSOLUTE(SEGM%,OFS%,REQLEN%,DMAIN%,STAT%,DMA2%)
IF STAT%=0 THEN RETURN
  PRINT "GPIB DMA  Input Error"
  IF (STAT%>0) THEN GOTO EndPgm
  PRINT "Warning: returned status ";STAT%
RETURN

```

WriteDMA:

```

-----
DMA OUTPUT
Writes un-formated (binary) character arrays
-----
OFS%=0 : STAT%=0 : SEGM%=-1 : STAT%=0
REQLEN%=INLEN%
OFS%=VARPTR(INDIGIT%(0))          ' Get addr of str descriptor,
CALL ABSOLUTE(SEGM%,OFS%,REQLEN%,DMAOUT%,STAT%,DMA2%)
IF STAT%=0 THEN RETURN
  PRINT "GPIB DMA  Output Error"
  IF (STAT%>0) THEN GOTO EndPgm
  PRINT "Warning: returned status ";STAT%
RETURN

```

# RECOVERY FROM ERRORS

The following alphabetical listing shows the error messages as they will appear on the 2045 display or a controller screen, followed by a brief suggestion for a remedy for the error condition and the most likely cause of the error, if the cause is not readily apparent in the message.

Keep in mind these messages should not always be limited to face value. Although you can expect the messages to be accurate responses to error conditions in most all cases, it is not possible to allow for all the actions that can cause an error. If, after thorough examination, an error message proves to be unfounded, other potential causes must be investigated.

Possible syntax errors can be identified in POLY mode without an attempt at computation. With R/S in STOP mode, pressing or sending ENTER will scan the polynomial for correct syntax and any perceived errors will be reported. The detection of evaluation errors such as 'value out of range' require R/S to be in RUN mode so the 2045 can attempt a computation.

Note: Where two similar messages are shown together and separated by a comma, this indicates either message may appear for a given cause.

## Error message and Cause / Recovery

### **A: Amplitude too high, set to 2.5V**

The requested amplitude from output A is beyond the 2045's capabilities. Make any necessary changes to time, math function results, modifiers, etc. to achieve a lower calculated amplitude of 2.5V peak or less.

### **A: Amplitude too low, set to 0V**

The 2045 will not accept negative amplitude values or positive amplitude values between 0V and 250uV (non-inclusive).

### **A: Offset too high, set to 3.5V**

The requested offset from output A is beyond the 2045's capabilities. Change the OFST modifier to achieve a lower effective offset of 3.5V or less.

### **A: Offset too low, set to -3.5V**

The requested offset from output A is beyond the 2045's capabilities. Change the OFST modifier to achieve a higher effective offset of -3.5V or more.

### **A: Requested NAMP not attainable**

Noise amplitude through the A output can be from 0 to 0.5V inclusive.

### **Amplitude (AMP) cannot be less than 0.**

Amplitude is given as peak value (positive magnitude from zero), hence cannot be negative.

### **ARCCOS undefined outside range [-1..1]**

Examine the length of the duration statement (FOR x, AT x, TO x) and the value or function inside the parentheses. If the result inside the parentheses goes past an absolute value of one in the time allotted to it, you need to shorten the time statement or change the math function or both.



**ARCSIN undefined outside range [– 1..1]**

Examine the length of the duration statement (FOR x, AT x, TO x) and the value or function inside the parentheses. If the result inside the parentheses goes past an absolute value of one in the time allotted to it, you need to shorten the time statement or change the math function or both.

**AT time cannot be < or = current time**

Change AT time to > current time.

**AT time must allow at least one point**

Increase AT time or change clock rate.

**B: Offset + Amplitude too high,  
clipped at + 0.5V**

Since the rated output of B is 1V p-p maximum, any peak extending beyond 0.5V will be clipped. Adjust amplitude/offset accordingly.

**B: Offset – Amplitude too low,  
clipped at – 0.5V**

Since the rated output of B is 1V p-p maximum, any peak extending beyond –0.5V will be clipped. Adjust amplitude/offset accordingly.

**B: Requested NAMP not attainable**

Noise amplitude through the B output can be from 0 to 0.1V inclusive.

**BAD OPEN - - OMEM or NVR**

RAM has been corrupted. Reset the 2045 or turn it off and back on. If message reappears, contact factory service.

**BLOCK size cannot be greater than 1000**

When performing an upload to an external device, define "BLOCK = n" where n is < or = 1000

**BLOCK size cannot be less than 0**

Define "BLOCK = n" where n is > or = 0

**Bus address must be 0 - 31**

Change bus address to within range.

**Can only enter an OP mode from either  
POLY or STANDARD FUNCTION mode.**

If the 2045 is already in OP modes 0, 4, 7, 232, 488, or 6000/6001, you must first send or press RESTORE before you can enter any other OP mode.

**Cannot divide by zero**

Check for division by a value or expression which equals zero.

**Cannot edit standard NOISE function**

The standard NOISE function can only be defined from its menu.

**Cannot evaluate polynomials in SFM.**

A polynomial sent remotely while in Standard Function mode cannot be evaluated in the SFM memory area. You must enter POLY mode first before sending the polynomial.

**Cannot take LN of a negative number**

Examine function to see if negative values exist and change sign if necessary.

**Cannot take LN of zero**

Check for any value or expression attached to LN which equals zero.

**Cannot take LOG of a negative number**

Examine function to see if negative values exist and change sign if necessary.

**Cannot take LOG of zero**

Check for any value or expression attached to LOG which equals zero.

**CLKLEV = 0 (0V) or 1 (1.5V)**

Set CLKLEV = 0 or 1 only.

**CLKSRC = 0 (internal) or 1 (external)**

Set CLKSRC = 0 or 1 only.

**D6000 data length error**

In OP 6000 mode, data from D6000 was not fully transferred. If record length is < 16K, retry. If length is > 16K, non-recoverable from 2045, need to shorten record in 6000.

**Device not ready**

Addressed device not ready to respond. Check device address and connections, retry.

**Device timeout**

Communications with device have not been established. Check bus address, device and connections, then retry.

**ECHO = 0 (OFF) or 1 (ON)**

Set ECHO = 0 or 1 only.

**EOIOUT = 0 (off) or 1 (on)**

Set EOIOUT = 0 or 1 only.

**Equation expected**

Change polynomial to equation form, e.g. [A = FOR 1m SIN(1K\*t)].

**ERRM = 0 (immediate) or 1 (on request)**

Set ERRM = 0 or 1 only.

**Error in parse tree structure - Pass 3**

Evaluation tree error has been detected. Check polynomial syntax.

**ERRSRQ = 0 (off) or 1 (on)**

Set ERRSRQ = 0 or 1 only.

**Exponent out of range**

Change exponent value.

**Expression evaluation error**

Check polynomial syntax.

**FASTIO = 0 (disabled) or 1 (enabled)**

Set FASTIO = 0 or 1 only.

**File already exists. Output segments cannot be overwritten.**

Check for duplication of two or more downloaded file names.

**File name not found, File not found**

Entered file name does not exist. Check file name against directory.

**Frequency (FREQ) out of range**

Frequency term must not be negative, equal to zero, or so small as to approach zero such as 1E-20.

**GET = 0 (none) or 1 (trigger) or 2 (sequence)**

Set GET = 0, 1, or 2 only.

**GPIB handshake error**

Communications error originating at controller. Check controller and retry.

**IEEE-488 GPIB comm. not installed.**

When attempting parallel IO, this message indicates that the IEEE chip and/or PAL on the CPU board are missing, not seated correctly, or faulty.

**Illegal modifier value**

Check for missing modifier value or change existing modifier value.

**Illegal character**

Look for non-standard character. Enter different character.

**Illegal clock (CLK) rate**

Check clock rate and change to within legal parameters.

**Illegal exponent**

When raising a base to a power, the exponent is not valid in form or value. Change accordingly.

**Illegal function argument**

Check entry, enter different argument.

**Illegal number in expression**

Verify correct form for file name (first character must be letter), check syntax, values, characters, retry.

**Illegal repeat value**

Verify repeat value is a whole number and not enclosed in parentheses.

**Illegal symbol**

Check syntax, non-standard symbols.

**Illegal waveform close: open loops**

All loops opened must be closed. Check for loops left open in remote data download command.

**Incompatible base and exponent values**

Check for problem such as negative value raised to real value. Edit as required.

**Incompatible offset due to amp.**

Change offset or amplitude value to bring within full scale range.

**Incorrect hardware configuration**

Bad hardware configuration data sent during DNLOAD/DATA download. Retry.

**Incorrect software revision level**

Waveforms uploaded from a certain revision machine should not be downloaded to a different revision machine.

**Infinite value encountered**

An overflow has occurred while parsing a number in an input string. Check for possible infinite or effectively infinite values. The maximum input value that can be parsed is  $5.76 \times 10^{18}$ .

**Insufficient memory, Insufficient memory available**

Insufficient space left in non-volatile memory to store more files. Increase free memory by deleting unneeded files.

**Insufficient memory for parsing**

Insufficient space for required evaluation tree. Polynomial expression is too complex and evaluation requires more memory than is currently available. Polynomial must be modified/simplified in some way.

**Insufficient memory to produce directory**

Open up some more memory by deleting any unneeded files.

**Insufficient number of points for CLK**

Increase duration of waveform or decrease clock rate to evaluate more points.

**Insufficient output memory available**

Decrease duration of waveform or increase clock rate to evaluate fewer points.

**Insufficient peak to peak amplitude**

Increase value of amplitude term.

**Insufficient space available in NVR**

Non-volatile RAM is full. Delete unneeded files from NVR.

**Insufficient TARGET PTS requested**

Increase TARGET PTS or change clock rate.

**Insufficient waveform data points**

Increase duration of waveform or decrease clock rate to evaluate more points.

**Integer value required**

Change entered value to integer.

**Integer values only for RPT**

Change RPT value to integer.

**Invalid GPIB bus address**

Verify correct bus address, change.

**Invalid entry, number expected**

Check for invalid characters and syntax.

**Invalid equation**

Check for correct equation syntax.

**Invalid filename**

See that first character is a letter, the name is  $<$  or  $=$  8 characters long and does not include "special characters" (such as colons, brackets, etc.). Filenames must start with a letter and be followed by numbers and/or letters only.

**Invalid function argument/expression**

Check syntax and values.

**Invalid function or expression**

Check syntax and values.

**Invalid modifier**

Check modifier syntax.

**Invalid modifier in MODIFIED mode.**

The waveform modifier you attempted to use has no meaning in modified mode. Check functional listing in the mnemonic reference section for applicability of modifiers.

**Invalid modifier in POLYNOMIAL mode.**

The waveform modifier you attempted to use has no meaning in polynomial mode. Check functional listing in the mnemonic reference section for applicability of modifiers.

**Invalid OMEM WRITE command**

Check for invalid data header in DIRECT DATA DOWNLOAD mode.

**Invalid operation requested**

Check for invalid OP code entry. If requesting a function such as OP488, make sure the appropriate chip and PAL are properly installed.

**Invalid STANDARD FUNCTION modifier.**

The waveform modifier you attempted to use has no meaning in standard function mode. Check functional listing in the mnemonic reference section for applicability of modifiers.

**Invalid time expression - use ( )s**

Check time expression for proper format. Time expressions such as (T + 1m) must be enclosed in parentheses.

**Invalid trigger mode**

Check for error in command code.

**IO Failure during xfer**

IO has failed during UPLOAD/DNLOAD. When using GPIB, check to see if EOI is being used correctly.

**KEY value must be an integer**

When sending KEY = n to "press" a key, n must be an integer from 0 to 63.

**LISNSRQ = 0 (off) or 1 (on)**

Set LISNSRQ = 0 or 1 only.

**Maximum LOW voltage limited by HIGH**

LOW cannot be higher than HIGH, change accordingly.

**Maximum number of output files exceeded**

Delete excess files.

**Memory allocation error. Insufficient memory available.**

Not enough memory for DNLOADed or DATA downloaded file. Open up some more memory or shorten file.

**Memory allocation error. Insufficient NVR available for segment p-block.**

Non-volatile RAM (nvr) is full. Delete excess files.

**Minimum HIGH voltage limited by LOW**

HIGH cannot be lower than LOW, change accordingly.

**Mismatched trigger signs**

When specifying +TO+ or -TO- trigger modes, check for consistency of signs, e.g. AT -TRIG TO -TRIG, not AT TRIG TO -TRIG.

**Missing (**

Missing left parenthesis is most likely syntax error. Count parentheses, add if # left does not = # right. Retry.

**Missing )**

Missing right parenthesis is most likely syntax error. Count parentheses, add if # left does not = # right. Retry.

**Missing ) in time expression**

Time expression requirements similar to voltage expression. Change accordingly.

**Missing = after equation name**

Add "=" symbol.

**Negative time values are not allowed**

Delete any minus signs in front of time values, or check for negative results of time expressions.

**No equation exists to be stored**

Enter or re-enter equation.

**No RS232 board installed**

When attempting to modify serial IO parameters via RS232, this message indicates the RS232 board has been removed, needs to be reseat/reinstalled due to poor connections, or may be faulty.

**No waveform data points calculated**

An attempt at calculation produced no data points. Some equations with unusable clock modifier values or frequency expressions may produce this condition. Rephrase the equation and recompute.

**No waveform exists to be stored**

Enter or re-enter waveform.

**No waveforms available.****Press <RESTORE> to exit.**

When using OP 6000, no waveforms have been recorded by a DATA 6000 prior to requesting a directory. Make sure the waveform file you want is listed in the D6000 directory, residing in the SYSTEM volume.

**Noise amplitude is limited by NBW**

The widest bandwidth, 354MHz, allows the highest amplitude.

**Noise amplitude (NAMP) out of range.**

Noise amplitude can be from 0 to 0.5V inclusive. Make sure NAMP does not have a negative value.

**Noise amplitude limited by signal amplitude**

If signal + noise > full scale range, decrease one or both.

**Noise SEED value cannot = 0**

NSEED value can be any integer value in the range +/-9999999 except NSEED cannot = 0

**Noise bandwidth (NBW) out of range.**

There are 15 bandwidths available for each possible noise clock rate (NCLK). Total range is from approx. 18nHz to approx. 354MHz. Make sure NBW is not negative.

**Nothing may follow a nested equation**

Delete any waveform segment coming after a nested equation.

**Nothing may precede a nested equation**

Delete any waveform segment coming before a nested equation.

**OMEM error: Invalid loop number**

During data download, call only loops 0, 1, or 2.

**OMEM error: Loop count out of range**

Count must not be < 0 or > 65535.

**OMEM error: Loop counter in use**

Loop must be closed before reopening.

**OMEM error: Loop not open at RPT close**

Loop must be opened before closing.

**Out of memory controller memory**

No more room for instructions. Decrease clock rate.

**Out of output memory**

No more room for data. Decrease clock rate.

**Output overvoltage or current limit**

Press R/S to disable, check equation for full scale range limits.

**OUTSEL = 0 (A) or 1 (B)**

Set OUTSEL (Output Selection) to 0 or 1 only.

**( or value or function expected**

Check syntax.

**Parse tree overflow**

Equation is too complex and requires more memory than is available to build a parse tree. Simplify.

**Port busy**

A port has been called for which is currently in use. Wait for operation to cease or abort operation if necessary.

**PROMPT = 0 (off) or 1 (on)**

Set PROMPT = 0 or 1 only.

**Repeats may only be nested 2 deep**

Delete any RPTs in excess of two nested per polynomial.

**Requested CLK rate too slow**

The polynomial cannot be computed at the requested clock rate. Enter a faster clock rate.

**Requested clock rate is not available**

Enter different clock rate.

**Requested frequency is out of range**

The frequency called for is either too low or too high. Increase or decrease as needed.

**Requested TARGET PNTS is too large**

Check OP 7 to see what memory is installed and/or enter smaller value.

**RPT count must be < 65536**

Enter smaller value.

**RPT count must be > 0**

Enter larger value.

**RS-232 communications not installed.**

When attempting serial IO, this message indicates the RS232 board has been removed, needs to be reseated/reinstalled due to poor connections, or may be faulty.



**Signal amplitude limited by noise amplitude,  
Signal amp. limited by noise amp.**

If signal + noise > full scale range, decrease one or both.

**Symbol length is too long,  
Symbol too long**

If > 8 characters, shorten it.

**TALKSRQ = 0 (off) or 1 (on)**

Set TALKSRQ = 0 or 1 only.

**TARGET PNTS must be > = 64 points**

Increase TARGET PNTS to > or = 64 points.

**TARGET PNTS must be specified as an integer**

Change to integer.

**This bandwidth is not available**

Change to available bandwidth.

**This is not an executable polynomial**

Check syntax.

**Time value out of range**

Increase or decrease as required.

**To run Tablet Draw, TARGET PTS cannot exceed 1000 points**

Enter OP 4 and check value of TARGET PTS is < or = 1000.

**TO time cannot be < or = to current time**

Increase TO time to > current time.

**TO time must allow at least one point**

Increase TO time or change clock rate.

**TRGLEV = 0 (0V) or 1 (1.5V)**

Set TRGLEV = 0 or 1 only.

**Unattainable clock (CLK) frequency**

Possible causes are  $CLK = f$  where  $f$  is too fast or too slow; Waveform period (in seconds)  $\div$  # of TARGET PTS forces unattainable clock rate; time duration expression is at or near zero or infinity.

**Unrecognized interface message**

GPIO interface message sent to the 2045 from an external device is bad.

**Upload abort: expecting -CONT-**

During an RS-232 upload, the 2045 will only recognize "CONT" as a command to send the next block of data. Sending anything else will abort the upload.

**Voltage level is too high**

Reduce voltage/amplitude term.

**Warning: Equation out of bounds, waveform clipped**

Requested amplitude of waveform exceeds 2045's capabilities. Redefine amplitude.

**Waveform delay (DLY) out of range**

Delay can be  $\pm 21.4748$  Sec. in 10nS or larger increments.

**Waveform phase (PHS) out of range**

Phase can be  $\pm 999999$ .

**Waveform period too short**

Increase duration of period.

**Waveform period too long**

Shorten period as required.



# SPECIFICATIONS FOR MODEL 2045

## WAVEFORM OUTPUTS:

- Low level analog waveform output
- High level analog waveform output with attenuator and offset

## DATA POINT UPDATE RATE:

- Internal or external data clock capability

## INTERNAL:

### MINIMUM CLOCK FREQUENCY:

- 0.0194Hz (515 secs/pt.)

### MAXIMUM CLOCK FREQUENCY:

- 800 MHz (1.25nS/pt.)

### ABSOLUTE ACCURACY:

- 0.005% @ clock frequency  $\leq$  100 MHz
- 0.025% @ clock frequency  $>$  100 MHz

### JITTER

- $<$  15pS RMS @ clock frequency  $>$  3.4 MHz
- $<$  75pS RMS @ clock frequency  $<$  3.4 MHz

## EXTERNAL CLOCK:

- D.C. to 250 MHz
- Ext. Clock threshold selectable as zero or 1.5 volts
- Slope is selectable as either Plus or Minus
- Maximum Safe Input Voltage  $\pm$  30 volts
- Minimum Input Voltage  $\pm$  500mV about selected threshold
- Ext. Clock Input Impedance 10 K $\Omega$ ,  $<$  50pf

## EXTERNAL ECL CLOCK:

- D.C. to 800 MHz
- ECL threshold only
- Input Impedance 50  $\Omega$  to -2V
- $V_{IH}$  = -0.8V to 0.0V
- $V_{IL}$  = -1.6V to -2.0V
- Max Input with no damage +0.5V, -4.0V

**HIGH LEVEL ANALOG OUTPUT (A):****MAX AMPLITUDE:**

- 5 volts p-p
- 10 volts p-p (open circuit)

**RESOLUTION:**

- 8 bits

**AMPLITUDE ACCURACY:**

- $\pm 3\%$

**ATTENUATOR RANGE:**

- 0 to 63dB

**ATTENUATOR ACCURACY:**

- $\pm 1\%$

**D.C. OFFSET:**

- User selectable, or internally computed;  
0 to  $\pm 3.5$  volts in 2mV steps with an accuracy of  
 $\pm 2\%$ ,  $\pm 10\text{mV} \pm (0.8\% \times \text{Waveform Peak-Peak})$

**NON-LINEARITY:**

- Differential non-linearity  $< 0.2\%$
- Integral non-linearity  $< 1\%$

**D.C. DRIFT:**

- $< \pm 2.0\text{mV/degrees C}$

**STEP RESPONSE:**

- Full scale rise and fall time (10% to 90%)  
 $< 2.2\text{nS}$  (DC Offset =  $0 \pm 2\text{V}$ )  
 $< 3.3\text{nS}$  (DC Offset =  $0 \pm 3.5\text{V}$ )
- Overshoot  $< 3\%$

**DATA POINT SETTling TIME:**

- $< 50\text{nS}$  to within 2% of final value with  
50 ohm resistive termination.

**AMPLITUDE FLATNESS:**

- $\pm 0.24\text{dB}$  from DC to 1 MHz
- $\pm 0.5\text{dB}$  from 1MHz to 10MHz
- $\pm 1\text{dB}$  from 10MHz to 30MHz
- $\pm 2.5\text{dB}$  from 30 MHz to 100MHz

**OUTPUT BANDWIDTH:**

- $> 160$  MHz, Typically 200 MHz

**OUTPUT LOW - PASS FILTERS:**

- Selectable, 2 or 20 MHz 3 pole Bessel

**SINEWAVE PURITY (HARMONIC AMPLITUDES):**

- < -40dBc from DC to 10 MHz
  - < -30dBc from 10 MHz to 30 MHz
  - < -20dBc from 30MHz to 100MHz
- (DC Offset =  $0 \pm 2V$ )  
(Add 10dBc for Offset > 2V)

**RESIDUAL NOISE:**

- < 0.1%  $V_{\text{OUTPUT p-p}} + 0.6\text{mV RMS}$
- < 0.2%  $V_{\text{OUTPUT p-p}} + 4\text{mV p-p}$

**DAC GLITCH ENERGY:**

- < 125 pv-sec. Effective width  $\approx 2.5\text{nS}$

**OUTPUT IMPEDANCE:**

- $50 \Omega \pm 10\%$

**OUTPUT PROTECTION:**

- Output will withstand short circuit to 0 Volts indefinitely

**LOW LEVEL ANALOG OUTPUT (B):****MAX AMPLITUDE:**

- 1 volt p-p
- 2 volts p-p (open circuit)

**AMPLITUDE VERNIER:**

- From 1V to 0.5V p-p in < 0.3mV steps

**OUTPUT COMPLIANCE:**

- With Externally Applied Offset =  $\pm 1.5V$

**RESOLUTION:**

- 8 bits with output = 0.5 to 1V p-p

**AMPLITUDE ACCURACY:**

- At 1V p-p output,  $\pm 4\%$

**D.C. OFFSET ERROR:**

- $\pm 3\text{mV}$

**NON-LINEARITY:**

- Differential non-linearity < 0.2%
- Integral non-linearity < 0.4%

**D.C. DRIFT:**

- <  $\pm 0.4\text{mV/degrees C}$

**STEP RESPONSE:**

- Full scale rise and fall time (10% to 90%) < 500 pS
- Overshoot < 7%

**DATA POINT SETTLING TIME:**

- < 7nS to within 1% of final value

**AMPLITUDE FLATNESS:**

- $\pm 0.2\text{dB}$  from DC to 10 MHz
- $\pm 1.0\text{dB}$  10 MHz to 100 MHz
- $\pm 2.0\text{dB}$  100 MHz to 200 MHz

**OUTPUT BANDWIDTH:**

- > 700 MHz, Typically 1 GHz

**SINEWAVE PURITY (HARMONIC AMPLITUDES & SPURIOUS SIGNALS):**

- < -46dBc from DC to 10 MHz  
(Spectra = 20 MHz to 200 MHz)
- < -36dBc from DC to 10 MHz  
(Spectra = 200 MHz to 1 GHz)
- < -25dBc from 10 MHz to 30 MHz
- < -15dBc from 30 MHz to 100 MHz

*Note: At higher frequencies aliased spectra are dominant. All readings are at Output = 1V p-p.*

**RESIDUAL NOISE:**

- < 1.5mV RMS
- < 6mV p-p

**DAC GLITCH ENERGY:**

- < 25pv-sec. Effective width  $\approx 0.5\text{nS}$

**OUTPUT IMPEDANCE:**

- $50\ \Omega \pm 10\%$

**OUTPUT PROTECTION:**

- Clamping Diodes at  $\pm 2\text{V}$ , maximum externally applied current =  $\pm 100\text{mA}$

**PROCESSORS:**

- Motorola 68000, 10 MHz, 32 bit (internal) with 256K byte EPROM program memory
- Intel 8031, 12 MHz, 8 bit with 16K byte EPROM program memory
- Motorola 68881 Math Processor
- Motorola 68440 DMA

**COMPUTATIONAL PRECISION:**

- All internal computations are done in Standard IEEE Floating Point with a 24 bit mantissa and an exponent range of  $\pm 10^{38}$ .

---

**DATA MEMORIES:**

- 512K data points

---

**OUTPUT MEMORY:**

- Is segmentable so that more than one waveform can reside in output memory. Output Memory is Volatile Static RAM.

---

**NON-VOLATILE MEMORY:**

- 78K bytes

Non-Volatile Memory is used to store "Polynomial" Waveform descriptions, by filename. Each filename can be up to 8 Alphanumeric characters and each file can contain up to 1280 characters. This memory is powered by lithium batteries which have a two year life with a 50% power-on duty cycle. Provision is made for field replacement with more conventional Alkaline batteries.

---

**MINIMUM WAVEFORM LENGTH:**

- 64 data points times the Clock Period

---

**HORIZONTAL RESOLUTION:**

- The total waveform must contain a multiple of 64 data points.

---

**EXTERNAL TRIGGER:**

---

**THRESHOLD:**

- Selectable as zero or 1.5 volt

---

**SLOPE:**

- Plus or Minus

---

**DELAY:**

- $< (40\text{nS} + 2 \text{ clock periods})$

---

**JITTER:**

- $< (100\text{pS} + 1 \text{ clock period})$

---

**INPUT IMPEDANCE:**

- $10\text{K } \Omega$  ( $< 50\text{pf}$ )

---

**MAXIMUM SAFE INPUT VOLTAGE:**

- $\pm 30$  volts peak

---

**MINIMUM INPUT VOLTAGE:**

- Plus and minus 500mV about selected threshold



**TRIGGER MODES:**

---

- Free Run, Manual (front panel button), Triggered Start, Triggered Stop, Gated by Trigger, and Triggered Start/Stop.

---

**MARKER OUTPUT:**

- Separate TTL output capable of driving 50  $\Omega$  load to ground
- Exhibits a POSITIVE transition at MARK Time of duration 32 data point clocks which yields one marker cycle per waveform duration. MARK Time is specified as an absolute time from the waveform start or, as a particular number of data point clocks from the start of the waveform.

---

**SYNC OUTPUT:**

- Separate TTL output capable of driving 50  $\Omega$  load to ground
- Exhibits a POSITIVE transition at the waveform start followed by a NEGATIVE transition 32 data points later.

---

**REMOTE INSTRUMENT CONTROL:**

- IEEE-488 and RS232 can be used to either control and program the instrument or to download direct waveform patterns.

---

**TRANSFER DATA RATES:**

---

**IEEE-488:**

- 300K bytes/sec using High Speed DMA

---

**RS232:**

- Up to 9600 baud

---

**WAVEFORM GENERATION MODES:**

- Mathematical (polynomial) notation by Front Panel entry or by remote communications
- Standard Functions by Front Panel entry or by remote communications.
- Standard Functions include Sine Wave, Square Wave, Pulse, Triangle, Sawtooth, and Random Noise.
- Direct Waveform Data Download via GPIB or RS232
- Captured Data reproduction with Data Precision Model 6000/6100 Waveform Analyzer
- Reproduction of Waveform generated by graphics tablet
- Machine setup Upload/Download via GPIB or RS-232

---

**GENERAL:**

---

**SPECIFICATION CONDITIONS:**

- 50 $\Omega$  resistive termination on both analog outputs, unless specified otherwise
- All outputs will double ( $\pm 5\%$ ) in amplitude when output load is open circuit, however some degradation in high frequency/speed performance can be expected.

---

**SPECIFICATION TEMPERATURE RANGE:**

- 0 to 40 degrees C

---

**OPERATING TEMPERATURE:**

- 0 to 40 degrees C

---

**POWER REQUIREMENTS:**

- 200 watts maximum
- 90-130 Vac or 180-260 Vac at 48-62 Hz

---

**DIMENSIONS:**

- 5.5"H X 16.5"W X 17" D  
(13.97 X 41.91 X 43.18 cm)

---

**WEIGHT:**

- 25 lbs. (11.3 kg)



**A**

amplitude, standard function D5  
 AT directive E6

**B**

batteries, non-volatile memory A4, B1  
 baud rate C3  
 bus address, IEEE-488 (GPIB)  
     OP488 C4  
     OP 6000 C5  
 bus address & command (CMD) device setting B7

**C**

clock  
     ECL B3  
     external B3  
     source C2  
     level C2  
 CLK modifier, poly mode E11  
 command (CMD) device  
     hardware setting B7  
     software setting C2  
 CYC/RAD mode E1

**D**

data block length restrictions E8, E9  
 delay D8  
 dip switch (figure B-3) B7  
 directory E3, E4  
 duty cycle D8

**E**

EOI C4, K3, K4, K5  
 error recovery section L

**F**

fast I/O C4  
 FILT modifier, poly mode E10  
 filter (FILT = ), standard function D6  
 filters, values for selecting G4  
 floating point values G7, G8  
 FOR directive E5  
 format, standard function menus D2  
 frequency, standard function D5  
 front panel A1, A2  
 fuse A3, A4, B1

**G**

GET (Group Execute Trigger) D3, D7

GPIB see IEEE-488

**H**

hardware/firmware version & special features see OP7

help B2

high/low level D9

**I**

IEEE-488 (GPIB)

bus address, setting B7

connector pin assignments B4

interface functions B4

OP488 C4

**K**

key code reference G21 to G23

**L**

listen SRQ C4

listen SRQ default setting B7

**M**

mark modifier, poly mode E10

marker out B3

marker points (MPTS:), standard function D6

marker time (MARK:), standard function D6

math functions

absolute value (ABS) E14

arccosine (ARCCOS) E13

arcsine (ARCSIN) E13

arctangent (ARCTAN) E13

cosine (COS) E13

exponent ( $Y^x$ ) E13

integral (INT) E13, E14

log, natural (LN) E14

log, common (LOG) E14

signum (SGN) E14

sine (SIN) E13

tangent (TAN) E13

math topics

math range E12

expression range E13

magnitude notation E13

parentheses nesting E13

special constants E13

trigonometric range E12  
 memory and file types E4  
 memory read/write test B2, H2, H14  
 modified mode menu fields F1, F2  
 modifiers, polynomial E9, E10, E11

## N

noise amplitude (NAMP) D8, H3, H14  
 noise bandwidth (NBW) D8, H3, H15  
 noise clock (NCLK) D8, H3, H15  
 noise mode (NOISE) D4, D5, D8, H3, H4, H15  
 noise seed (NSEED) D8, H3, H15

## O

offset (OFST) modifier E10  
 operation (OP) codes  
   factory diagnostic C7  
   OP0 C1, section I, H2, H16  
   OP4 C1, C2, H2, H16  
   OP7 C2, C3, H2, H16  
   OP232 C3, H2, H16  
   OP488 C4, H2, H17  
   OP1510 C5, section J, H2, H17  
   OP 6000 C5, C6, H2, H17  
   OP 6001 C6, H2, H17  
   OP 9999 C7, H2, H17  
 output, selecting signal C1, C2

## P

period D9  
 phase D8  
 polynomial (arbitrary waveform) mode  
   directive statements E5  
   editing E2  
   entering E1  
   expressions and directory size E3  
   mixed directives E6  
   modifiers E9, E10, E11  
   store, recall, delete E3  
   time expressions E5  
   triggering E7, E8  
   waveform examples E15 to E23  
 port  
   serial (RS-232) B5, B6, C3  
   parallel (IEEE-488) B4, C4  
 power source B1  
 pulse width D9

**R**

- rear panel A1, A3
- remote mnemonics section H
  - commands H2
  - functional listing H1 to H4
  - general reference H5 to H24
  - math operators H4
  - modifiers H3, H4
- remote programming and control section G
  - direct data download G2
    - closing data header G6
    - data G6
    - data header G5
    - file construction G3 to G6
    - file types G2
    - IEEE floating point numerical conversions G7, G8
    - main header G3
    - remote control example G1, G2
    - sample programs G8 to G17

RESTORE C1, C3, C6, C7, F1, H2, H20

RPT statement

- restrictions E7
- trigger considerations when using E8

RS-232

- ports B5, B6
- connector pin assignments B6
- parameters C3
- selecting C3

**S**

- scope draw C1, section I
- setting default command (CMD) device B7
- setting default GPIB bus address B7
- setting default listen SRQ mode B7
- signal out connectors B3
- sine wave mode D3
- S + N modifier E9, E10
- square wave mode D3
- SRQ
  - setting listen B7
  - operation G18 to G20
- status bits G18
- swept waveforms, generating E11, E12
- symmetry D9
- sync out B3

**T**

T (global time) vs t (local time) E5  
tablet draw (OP1510) C5, section J  
talk SRQ G18 to G20  
target points C2  
time expressions E5  
TO directive E5, E6  
triangle wave mode D3  
trig level C2  
trigger mode D6, D7, G4  
trigger in B3  
triggering, poly mode E7, E8

**U**

upload/download E4, section K

**V**

voltage select card removal A3, B1





## **ORDERING INFORMATION**

### **ORDERING ADDRESS:**

ANALOGIC Corporation, Electronic Products Group, 8 Centennial Drive,  
Peabody, MA 01961  
Telephone: 508-977-3000, 800-451-1816; Telex: 6817144; Fax: 508-532-6097,  
508-531-1266

### **QUOTATIONS:**

Upon request, quotations and pro forma invoices can be furnished to you.

### **NET 30 DAYS**

ANALOGIC's standard terms of sale are NET 30 days following the date of shipment. As with all credit items, satisfactory credit accommodations must be arranged.

### **SHIPMENT**

All prices, quotations and shipments are FOB Peabody, MA unless otherwise specified.

Unless otherwise specified, shipment will be made via the most economical method. Surface and air shipments will be insured at full valuation unless your order instructs otherwise.

### **U.S. GOVERNMENT SALES**

Some of the products in the price schedule are listed on G.S.A. federal supply schedule contracts. Please contact ANALOGIC for more information.

All instruments made in U.S.A. (Factory) Peabody, MA

Prices and specifications subject to change without notice.

# **ANALOGIC**

 **DATA PRECISION**

- ANALOGIC Corporation, 8 Centennial Drive, Peabody, MA 01961 USA  
Tel: 508-977-3000 Telex: 6817144 Fax: 508-532-6097, 508-531-1266
- ANALOGIC GmbH Siemensstrasse 13, 6200 Wiesbaden-Nordenstadt, West Germany  
Tel: 49 6122 70060 Teletex: 6122969 ANALO WI Fax: 49 61 2215262
- ANALOGIC Ltd., Ascot House, Doncastle Road, Bracknell, Berkshire, England RG12 4PE  
Tel: 44 3 44 860111 Telex: 847145 ANALOG G Fax: 44 3 44 860478